NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

TRACKING HUMAN WALKING USING MARG SENSORS

by

Ioannis Pantazis

June 2005

Thesis Advisor:	Xiaoping Yun
Second Reader:	David C. Jenn

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2005	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Tracking Human Walking Using MARG Sensors			**5. FUNDING NUMBERS**
6. AUTHOR(S) Ioannis Pantazis			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Office (ARO) U.S. Navy Modeling and Simulation Office (N6M)			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

This thesis addresses modeling and simulation of the human lower extremities in order to track walking motion and estimate walking distance. The lower extremities are modeled as an articulated object, which consists of rigid bars connected to each other by joints.

This model is tested by using both synthetic and real data. The synthetic data is created based on the main principles of biomechanics. The real data is obtained from the MARG sensors and is processed by the Factored Quaternion algorithm. Next, it is implemented in a simulation program written in Matlab. The program utilizes a mathematical model that represents the human gait–cycle and is based on the theory of forward kinematics as well as on the theory of manipulator kinematics.

The simulation program is able to track the motion of the limbs that represent the lower extremities and estimate the traveled distance. Extensive laboratory tests verified the validity of the configuration.

14. SUBJECT TERMS MARG Sensors, Gait–Cycle, Forward Kinematics, Manipulator Kinematics, Human Walking Simulation, Position Estimation			**15. NUMBER OF PAGES** 117
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

TRACKING HUMAN WALKING USING MARG SENSORS

Ioannis Pantazis
Lieutenant Junior Grade, Hellenic Navy
B.S., Mechanical Engineering, Hellenic Naval Academy, 1998
B.S., Equivalency in Electrical Engineering, Naval Postgraduate School, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
and
MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2005

Author: Ioannis Pantazis

Approved by: Xiaoping Yun
Thesis Advisor

David C. Jenn
Second Reader

John P. Powers
Chairman, Department of Electrical and Computer Engineering

Dan C. Boger
Chairman, Department of Information Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis addresses modeling and simulation of the human lower extremities in order to track walking motion and estimate walking distance. The lower extremities are modeled as an articulated object, which consists of rigid bars connected to each other by joints.

This model is tested by using both synthetic and real data. The synthetic data is created based on the main principles of biomechanics. The real data is obtained from the MARG sensors and is processed by the Factored Quaternion algorithm. Next, it is implemented in a simulation program written in Matlab. The program utilizes a mathematical model that represents the human gait–cycle and is based on the theory of forward kinematics as well as on the theory of manipulator kinematics.

The simulation program is able to track the motion of the limbs that represent the lower extremities and estimate the traveled distance. Extensive laboratory tests verified the validity of the configuration.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my thesis advisor Professor Xiaoping Yun for giving me the opportunity to work on this great project. Professor Yun was my teacher in several courses before he became my thesis advisor and I had the chance to appreciate his communicability of knowledge and his ability to make difficult concepts clear to his students. During the research and writing of my thesis, I was impressed by the way he managed to resolve problems that appeared and by his commitment to providing all necessary help.

I would also like to thank Professor Robert B. McGhee for passing me large fragments of his knowledge in human motion and in LISP, as well as for the LISP program in the human gait–cycle that he provided.

I strongly appreciate all the help that James Calusdian provided me in the lab. He was always there when I needed his help.

I would like to thank my brother–in–law Stefanos Filtikakis not only for his excellent job in helping me with the measurements, but also for being my best friend during the last two years.

Special thanks to Andreas Kavousanos–Kavousanakis for his valuable recommendations even from such a long distance.

I cannot find words to express my love and gratitude to my parents Panayiotis and Maria Pantazis and my sister Eleni for their love and support.

Finally, I would like to dedicate this work to my beloved wife Dimitra Christopoulou who for the last six years has been my inspiration and the reason for trying hard to be a better person. My only concern in life is how to make her happy.

THIS PAGE INTENTIONALLY LEFT BLANK

Στην Ντίμη μου

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

This thesis aims to introduce a new chapter in the magnetic, angular rate, and gravity (MARG) sensor project. The objective of the MARG sensor project is to track human motion, based on inertial/magnetic sensors, which measure the orientation of the human limbs. So far the MARG sensor project dealt only with tracking the motion of the right hand. However, in order to implement this entire project to build motion–tracking body suits or to create a virtual environment, mainly for training purposes and other military applications, all the other limbs of the human body must participate as well.

The main objective of this thesis was to utilize the MARG sensors in order to track the motion of the lower extremities and estimate the position of the model that executes the movement. A simulation program was created in Matlab in order to accomplish this.

For the purposes of this research, a MARG III sensor was used. The sensor is attached to a limb and data is obtained as the limb moves. The MARG III sensors consist of three accelerometers, three magnetometers, and three angular rate sensors orthogonally mounted. Thus, data is derived from nine sensor elements and, in order to be wirelessly transmitted to a central terminal, it must to be sent through a single channel. Therefore, it has to be multiplexed. This is the task of a control interface unit (CIU). This thesis uses a three–channel CIU.

The data is further processed by the *Factored Quaternion* algorithm, which transforms the "raw" data delivered from the sensors into quaternions and then into angles. These angles represent the orientation of the limb in all directions. Next, the sets of angles are provided to a simulation program, which implements human walking.

Initially, the simulation program was driven by pseudo–data, which represented small increments or decrements of the angles of the limbs. In order to construct this pseudo–data, an in–depth research on the fundamental biomechanics of human walking and, especially, of human gait–cycle was executed.

The simulation model implemented in a Matlab program is based on the theory of forward kinematics, as well as on the theory of manipulator kinematics due to the simplicity and the low computational load they ensure.

The results of the simulation using real data were satisfactory since the program was able to track the motion of the lower extremities and to estimate the position of the model with desired accuracy.

I. INTRODUCTION

This chapter discusses the previous research conducted on the MARG sensors and, more specifically, on their implementation in tracking human motion. Next, the thesis goals are presented along with some other critical issues that must be addressed. Finally, a brief discussion of the outline of the thesis chapters follows.

A. MOTIVATION AND PREVIOUS RESEARCH

This thesis is part of the MARG project. The MARG project is an on–going effort to track human motion to create a virtual environment. This is accomplished by utilizing magnetic, angular rate, and gravity (MARG) sensors, which are inertial/magnetic sensors. One sensor is attached to one limb and provides data according to the motion of the limb. Data from the sensor is processed via a control interface unit (CIU) and an algorithm calculates the orientation of the limbs. As it can be easily understood, such a project can be applicable to virtual combat training programs. The size of the sensors, which is considerably small, allows them to be attached to a full body suit. With the appropriate auxiliary hardware and compatible software, tracking the motion of every limb is possible.

Eric R. Bachmann [Ref. 1] attempted to create a synthetic environment by utilizing the MARG II sensors. One sensor was attached to each limb and data from all sensors was delivered to a terminal, where it was processed to depict the motion of the limbs on a human avatar (Figure 1). However, this configuration did not come without drawbacks. The considerably large dimensions and the weight of the equipment as well as high power consumption were problems that had to be resolved. Furthermore the data transmission was not wireless. Andreas Kavousanos-Kavousanakis [Ref. 2] presented a different configuration, in which data transmission was wireless and the dimensions of the equipment along with the power consumption were significantly reduced. Furthermore, the *Quest* algorithm was implemented to calculate the orientation of the limbs. The algorithm used the properties of the quaternions and was able to track the motion of the right arm in real time, by utilizing two MARG III sensors (one at each limb). However, the *Quest* algorithm proved to be inadequate when large linear acceleration occurred. In order to address that problem, Conrado Aparicio [Ref. 3] designed a Kalman filter and was

able to track the motion of the right arm efficiently in real time even when large linear accelerations occurred. The MARG III sensors were also utilized in this case.

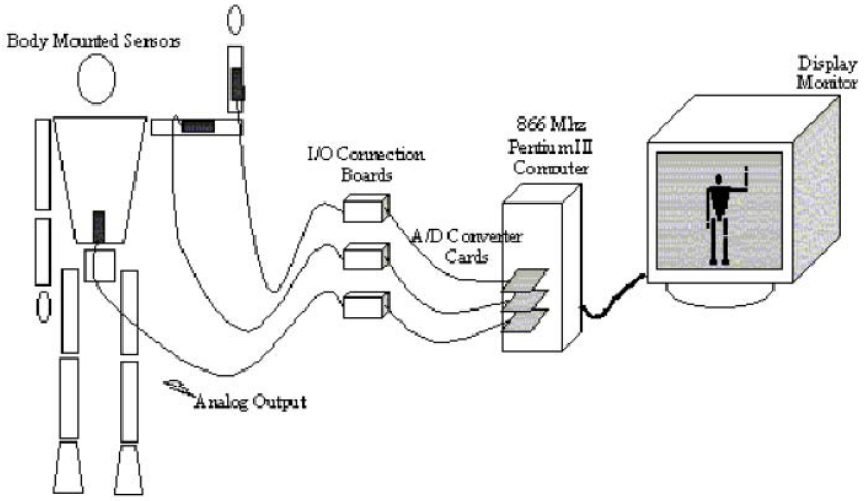


Figure 1. Initial MARG Project Configuration [After Ref. 4.]

This thesis attempts to move the MARG project to another level by studying the motion of the lower extremities. The MARG III sensors are used so that data can be derived and transmitted via the CIU to a central terminal. Then, the data is processed by the Kalman filter and imported into a Matlab program, in order to track the motion of the lower extremities. The first version of the Matlab program consisted of pseudo–data, which was generated based on the basic principles of biomechanics. In addition, a mathematical model, based on the theory of forward kinematics and on the theory of manipulator kinematics, was implemented.

B. THESIS GOALS

The main goal of this thesis was to develop a Matlab simulation program that models the lower extremities as an articulated object, tracks their motion, and estimates the traveled distance of the model. In order to achieve this goal, it was necessary to address the following:

- Provide a description of walking motion and analyze the human gait–cycle,
- Derive a mathematical model to describe human walking, using forward kinematics and manipulator kinematics,
- Implement the mathematical model in a computer simulation in order to describe the full gait–cycle,
- Use the MARG III sensors to provide real data,
- Process the data and implement it in a Matlab program in order to track the motion of the lower extremities and to estimate the traveled distance of the model.

C. THESIS ORGANIZATION OUTLINE

Chapter II provides a theoretical background on human body locomotion and discusses the patterns of human walking. Furthermore, it analyzes the human gait–cycle and discusses certain attributes, such as, the location of motion, the direction of motion, the determinants of gait, and other crucial parameters.

Chapter III provides a discussion of kinematics and dynamics. The goal of this chapter is to derive a mathematical model in order to describe human walking adequately. Forward kinematics theory was chosen for this purpose mostly due to its simplicity. Furthermore, in order to model the lower part of the human body, a discussion in manipulator kinematics is provided.

Chapter IV provides a computer simulation of human walking by implementing the mathematical model presented in Chapter III, in a program written in Matlab. In addition, this chapter discusses several past approaches of simulating human walking.

Chapter V presents a more advanced and sophisticated model, which represents the lower extremities, and is implemented with a program written in Matlab to simulate the human gait–cycle more effectively. In addition, a program in LISP is illustrated in order to describe the behavior of the foot during the human gait–cycle. Finally, a brief comparison between the two approaches is provided.

Chapter VI discusses the concept of tracking human motion and the concept of position estimation. The implementation of the MARG sensors to derive real data is discussed. Furthermore, a brief description of the equipment is provided. This chapter also

discusses the processing of the data and how it was implemented in the Matlab simulation program to move the model and to estimate its position. Finally, the results of this procedure are presented.

The final chapter presents the conclusions derived from this research as well as some suggestions regarding the continuation of this project and recommendations for future work.

Appendix A contains a Matlab program which tests the mathematical model of the human–gait that is discussed in Chapter III. Appendix B contains a Matlab simulation program for the case which is presented in Chapter IV. Appendix C contains a Matlab simulation program with the enhancements presented in Chapter V. Appendix D contains a LISP program written by Professor Robert McGhee that is discussed in Chapter V. Appendix E presents a Matlab simulation program that processes real data. This program is discussed in Chapter VI.

II. DESCRIPTION OF MOTION–GAIT ANALYSIS

This chapter discusses the walking model of the human body and provides a description of walking motion, which contains a brief illustration of the types of motion, the location of motion, and the direction of motion. The main part of this chapter provides a thorough description of human locomotion, and analyzes the gait–cycle along with some other crucial parameters. The end of the chapter presents another group of components of gait, called determinants of gait.

A. TYPES OF MOTION

The following material is mainly taken from [Ref. 5].

The human skeleton could be described as a system which contains rigid bars and joints. Rigid bars have different lengths and sizes, depending on which bone they represent, connecting to each other with joints. Describing the human body is not an easy task. One leg alone has 29 bones and 37 muscles [Ref. 6]. However, the objective of this thesis is not to give a precise description of the biomechanics function of the lower part of the body, but to illustrate a model used in future chapters to derive the human walking model.

There are four types of motion that describe every possible move of which an object is capable. The angular motion is the motion of an object around an axis, with the object having a constant distance from the axis of rotation. Each point of the object moves through the same angle, at the same time. As seen in later chapters, although joints present a rather complex movement, their movement can be considered purely angular for reasons of simplicity.

The second type of motion is called translatory or linear motion. In this kind of movement, the object travels in a straight line. In other words, all parts of the object travel parallel to each other and have the same distance at the same time.

The combination of the angular and translatory motion gives the third type of motion, which is called curvilinear motion. This occurs when an object is traveling from one point to another while at the same time it is rotating around an axis. A classical example

is the Earth, which is traveling into space while it is rotating around its own axis. However, this is more complicated since the Earth rotates around the sun, and our solar system and galaxy travel through space as well.

The fourth and last type of motion is called general plane motion. General plane motion is very similar to the curvilinear motion. The difference is that the object is segmented and free to move. Walking, as well as most of the movements of the limbs of a human being, belongs to the general plane motion category. For example, the movement of the lower limb of a leg contains a rotation and a translation and has no fixed way of traveling. The rotation of the limb takes place around the vertical axis, which goes through the joint that connects the lower with the upper limb of the leg. Translatory movement of the limb occurs when it travels (forwards or backwards, upwards or downwards) from one position to another during the gait–cycle. This kind of movement is not fixed, although there are certain limitations to be discussed later. On the other hand, there are certain parts of the body that maintain their linear identity during walking. The head is a characteristic example of translatory movement during walking.

1. Location of Motion

A proper description of the movement of an object should include the place or the plane where the movement occurs. Given that the three–dimensional coordinate system is used, the motion of the object could take place in the horizontal, frontal or sagittal plane. For clarification, those planes with respect to a person standing in an anatomic position are defined[1]. However, it is first necessary to define the coordinate system. The positive *Z*–axis has the forward direction of the standing person, the positive *Y*–axis goes up and the positive *X*–axis is found according to the right–hand rule. The horizontal plane corresponds to the *X*–*Z* plane, and divides the body into lower and upper halves as shown in Figure 2a. The frontal plane corresponds to the *X*–*Y* plane, and divides the body into front and back halves as shown in Figure 2b. The sagittal plane corresponds to the *Y*–*Z* plane, and divides the body into right and left halves as shown in Figure 2c. When an object is moving in one of these planes, it is either rotating around its axis or translating from one point to another, with a path parallel to the plane or both. Of course, it is not possible to restrict human movements in that manner.

[1] An anatomic position is the position where the person stands, looking forward.

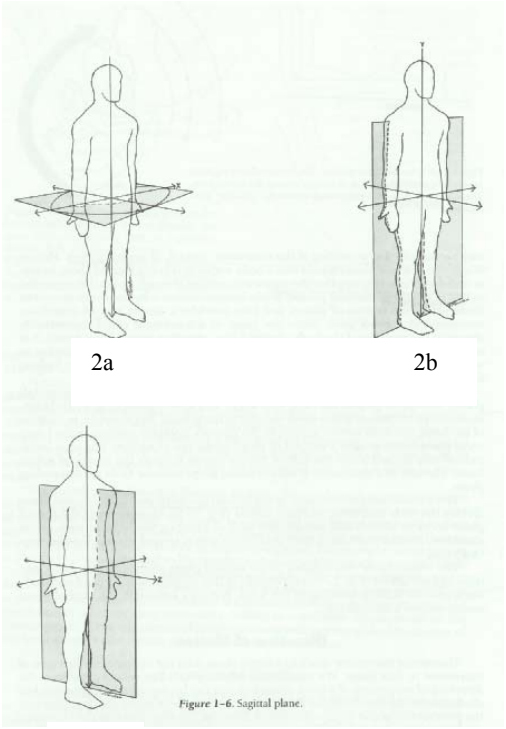


Figure 2. Horizontal, Frontal and Sagittal Plane [From Ref. 5.]

2. Direction and Quantity of Motion

Descriptions of the direction of movements depend on the nature of the movement itself. In the case of translatory movement, signs are given conventionally. Up and right motions take positive values, and down and left motions take negative values.

For rotatory motions, signs are set arbitrarily to clockwise and counterclockwise. Based on the plane on which the motion occurs, the following categorization applies:

- Flexion and extension occur in the sagittal plane. According to Norkin and Levangie, "Flexion refers to rotation of one or both bony levers around a joint axis so that ventral surfaces are being approximated" [Ref. 5]. When the movement occurs in the same plane with a different direction, it is called extension.
- Abduction, adduction, and lateral flexion mainly occur in the frontal plane. According to Norkin and Levangie, "Abduction is rotation of one or both segments of a joint around an axis so that the distal segment moves away from the midline of the body" [Ref. 5]. The movement that occurs in the same plane in an opposite direction is called adduction. Lateral flexion occurs when the moving segment is part of the midline of the body.

- Medial and lateral rotations occur in the horizontal plane of the body. By quoting Norkin and Levangie again, "Medial rotation refers to rotation towards the body's midline, and lateral rotation refers to the opposite motion" [Ref. 5].

Note that for the purposes of this thesis, movement in the sagittal plane, and therefore flexion along with extension, is the main field of development of the human walking model. Representing the human full gait–cycle in the plane involves only rotation of the limbs in a sagittal plane around a frontal axis.

Rotatory motion is measured either in degrees or in radians. As is known, 360 degrees correspond to 2π radians (approximately 6.28 radians), and one radian corresponds to 57.3 degrees. In translatory motion, the distance that the object travels (displacement) is being measured. Units of meters will be used.

B. GAIT–CYCLE

The following material is mainly taken from [Ref. 7].

Human gait has been a field of interest and research since prehistoric times. From the cave drawings of Cro–Magnon to the sculptures and the athletic activities depicted on the pottery art works in ancient Greece (Figure 3), people seemed to have a particular interest in the patterns of movement of humans and animals. Of course, from a scientific point of view, gait science has been rapidly developing during the past decades. Due to recent progress, scientists currently are capable of recording and analyzing all kinds of movements from the gait–cycle of a child to the performance of an athlete [Ref. 8].

The main characteristic of human gait is that, just like voice or fingerprints, it is almost unique for every person. One may claim that, in general terms, the gait–cycle of a person is similar to the gait–cycle of every other individual, which is exactly the point. It may be similar but it is not exactly the same. Every person has a unique characteristic gait pattern. Gait pattern varies, depending on the health status, personality, occupation, age, sex and many other attributes. Some characteristic gait patterns are those of a soccer player, a military person, or a sailor. Apart from the individualistic nature of the human gait, each of us adjusts the gait to the various circumstances of every day life. For example, sometimes a defensive gait pattern is being developed. Other times, it is an offensive one or a gait pattern developed when someone is in a hurry.

Figure 3. Prehistoric and Ancient Depictions of Human and Animal Locomotion [After Refs. 9 and 10.]

The reason for the different kinds of gait patterns is that gait depends on certain attributes such as the construction of the bones, the position and function of neuromuscular structures, and the function of the joints. However, the detailed explanation of such operation is beyond the scope of this paper. The objective is to illustrate how human locomotion is achieved in terms of the gait–cycle.

When someone is walking, the overall movement observed from the outside is purely a translatory movement. This translation of the body takes place through rotatory movements of the lower limbs, which in a healthy person, are in complete harmony with each other. Therefore, the lower extremities or, in other words, the feet, the legs, and the thighs are cooperating with each other in order to move the HAT (head, arms and trunk). The term gait–cycle refers to the period of time between two identical positions of the same lower extremity.

1. Phases of the Gait–Cycle

In order to be able to understand gait better, scientists divided human walking into many parts. One of the pioneers was an American prosthesist, A. A. Mark, who more than a century ago, presented medical society with an analysis, in which he divided the gait in eight contiguous phases. He also implemented kinetoscopic photography in order to present a qualitative analysis (Figure 4). In 1885, a French physiologist named A. Marey used a method similar to Mark's to depict displacements in human gait.

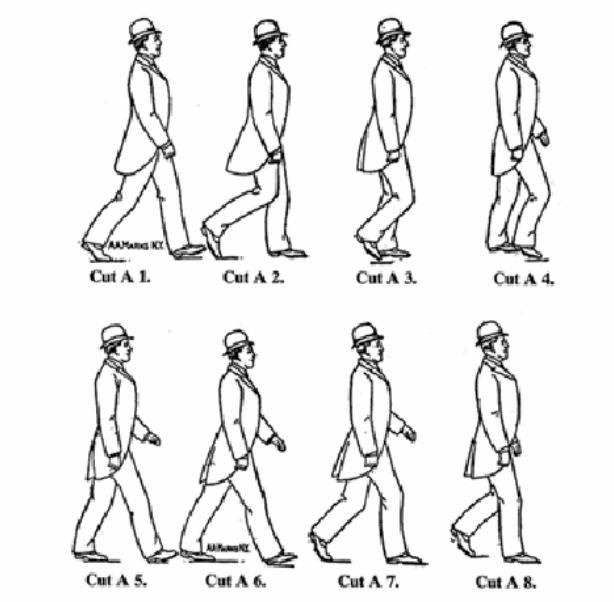


Figure 4. The Eight Contiguous Phases of the Gait According to A. A. Mark [From Ref. 7.]

Since then, thousands of papers have been written and countless presentations given, trying to provide more insight into this subject. Different perspectives gave different interpretations on the manner in which the gait should be divided. Depending on whether the author was an orthopedic, a physician or a mathematician, gait was defined in various ways. However, for the sake of consistency, only one description of gait–cycle is presented in this thesis. According to the professional staff at Rancho Los Amigos Medical Center, in California, walking involves three main tasks [Ref. 11]:

- Weight acceptance
- Single–limb support
- Swing limb advancement

The gait–cycle starts at the point of initial contact and ends when the very same contact occurs again. During the gait–cycle, extremities pass through two phases, a single stance phase and a single swing phase [Ref. 5]. The stance phase occurs for as long as the reference lower extremity is in contact with the ground. It starts at the initial contact of the extremity, which is when the heel touches the ground, and it is completed when the toe leaves the ground. The stance phase comprises 62 percent of the gait–cycle.

The swing phase refers to the part of the cycle in which the same extremity does not touch the ground. It starts right after the moment the toe leaves the ground and it is completed right before the moment of impact between the heel and the ground. The swing phase encompasses approximately 38 percent of the gait–cycle.

Another important period of human locomotion is the double support, which occurs when both lower extremities are in contact with the ground at the same time. There are two double support periods in one cycle. Those two periods of double–limb support represent 25 percent of the gait–cycle. Figure 5 illustrates the phases of the gait–cycle.

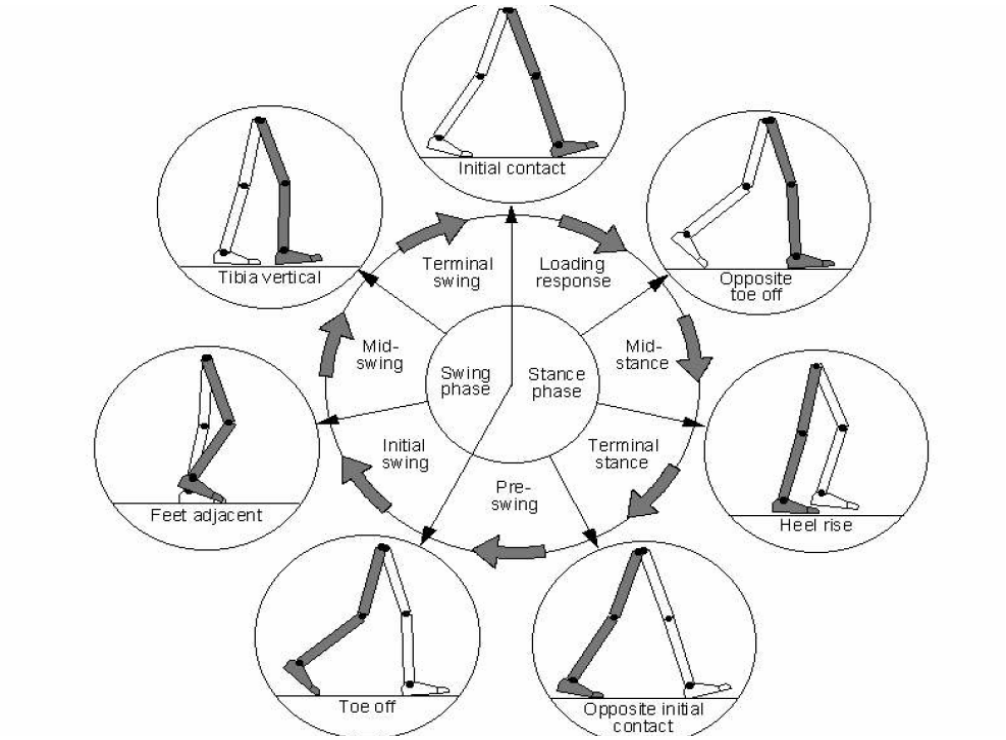


Figure 5. Phases and Subdivisions of Gait–cycle [From Ref. 12.]

2. Subdivisions

Stance is further divided into five subunits: heel strike, foot flat, midstance, heel off, and toe off. Heel strike refers to the moment that the heel strikes the ground. Foot flat refers to the period that the foot is completely attached to the ground. Midstance is the moment where the HAT is approximately aligned to the supporting extremity. Heel off is the moment where the heel loses contact with the ground. Finally, toe off refers to the

moment where the toe is in contact with the ground. Note that a reference extremity has been defined (either right or left).

Swing is divided into three subunits: acceleration, midswing, and deceleration. Acceleration begins immediately after toe off of the reference and is completed when the HAT is aligned to the other extremity. Midswing occurs when the reference extremity passes beneath the HAT. Deceleration begins right after midswing. During this period, the knee is extending and is ready for heel strike. It ends right before the heel strike, which initiates a new gait–cycle.

The professional staff at the Rancho Los Amigos (RLA) Medical Center has recently presented another set describing the subunits has been presented recently, the RLA method. This method describes the gait and, unlike the traditional terminology, it refers to lengths of time, giving a more adequate definition of the starting and ending points of a subunit. Table 1 compares traditional and RLA terminology. In addition, Figure 6 illustrates the phases of the gait–cycle along with its subdivisions in both traditional and RLA terminology.

Traditional	**RLA**
Heel Strike	Initial Contact
Heel Strike to Foot Flat	Loading Response
Foot Flat to Midstance	Midstance
Midstance to Heel off	Terminal Stance
Toe off	Preswing
Toe off to Acceleration	Initial Swing
Acceleration to Midswing	Midswing
Midswing to Deceleration	Terminal Swing

Table 1. Comparison between Traditional and RLA [After Ref. 5.]

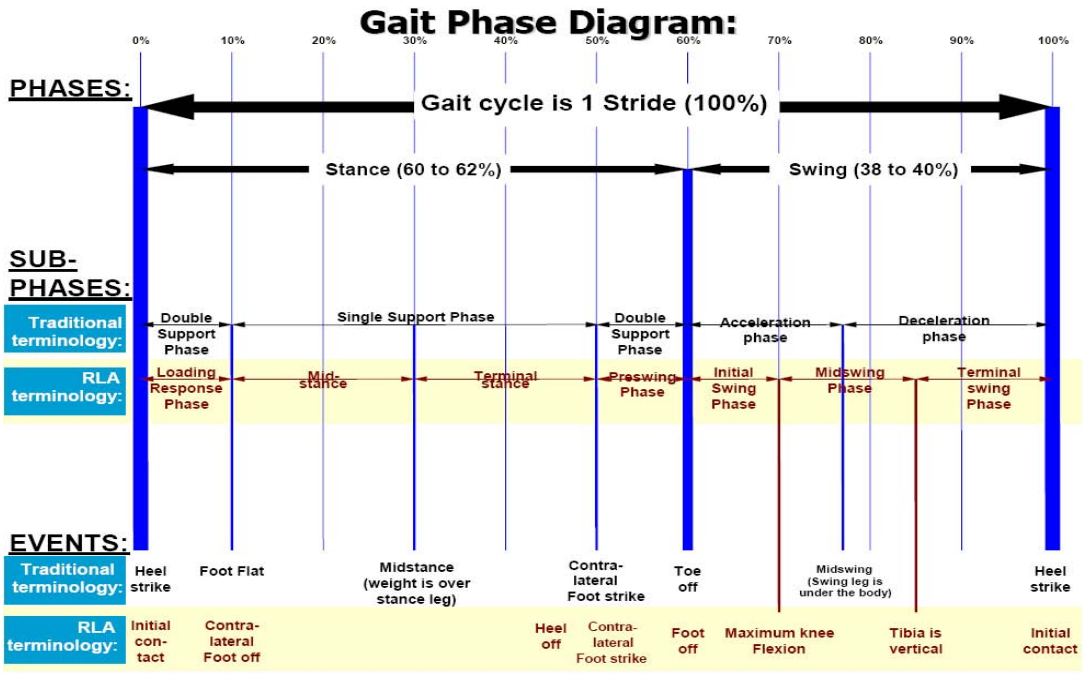


Figure 6. Gait Phase Diagram [From Ref. 12.]

3. Time and Distance Parameters of Motion

The following material is mainly taken from [Ref. 5].

This section discusses several other characteristics of human locomotion.

The time–related characteristics of the gait are called temporal variables. Such variables are: stance time, single–limb time, double–support time, swing time, stride and step time, cadence, and speed. Stance time is the period of time in which the stance phase occurs. Single–limb time is the period of time in which HAT is supported by one extremity. Double–support time is the period of time in which both lower extremities are in contact with the ground. Stride time is synonymous to gait–cycle duration. Step time is the total amount of time required for one step. Cadence is the number of steps per unit of time (steps per second or steps per minute). Speed is referred to as free, slow and fast. Free speed refers to a person's normal and undistracted walking speed, while slow and fast speeds are rather self-explanatory terms.

The distance related parameters of motion are called distance variables and include stride length, step length, width of walking base, and degree of toe out. Stride length is the linear distance from one heel strike to the next for the same lower extremity.

Stride length increases as the speed increases [Ref. 13]. Step length is the linear distance from one heel strike of one extremity to the next heel strike of the opposite extremity. The width of the base support is the linear distance between one point of the heel of one lower extremity, and the same point of the heel of the opposite extremity. Degree of toe out is the angle of foot placement.

C. DETERMINANTS OF GAIT

The following material is mainly taken from [Ref. 7].

Another group of components being considered quite often in kinematics analysis is called determinants of gait. During walking, the center of gravity (COG) of the body makes a translatory movement. The determinants of gait are trying to maintain the movement of the COG to a minimum. There are six determinants: variations in pelvic rotation, pelvic tilt, knee flexion at midstance, foot and ankle motion, knee motion, and lateral pelvic displacement.

There are forward and backward pelvic rotations which directly relate to the forward and backward movements of the lower extremities. The total range of pelvic rotation is eight degrees, four for the forward and four for the backward rotation. Figure 7 illustrates the pelvic rotation in the transverse plane.

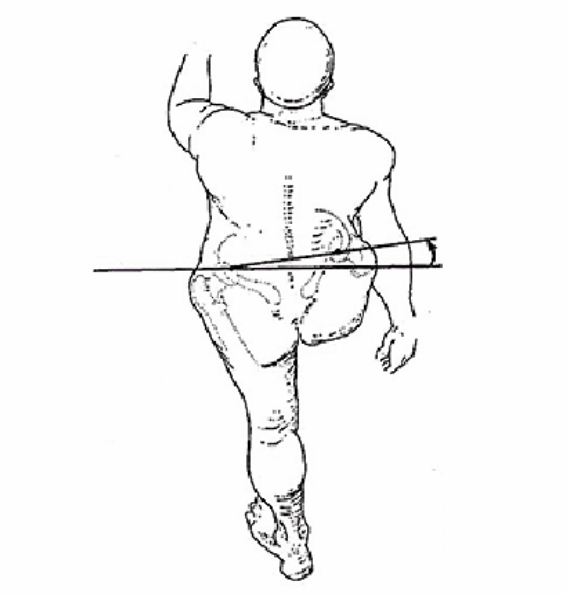


Figure 7. Pelvic Rotation [From Ref. 7.]

The COG reaches its highest point at midstance. Note that the trajectory of the COG in walking has a sinusoidal form. This highest point would be even higher if it was not for the pelvic tilt to depress the COG. The total range of the tilt towards the swing side is five degrees from vertical. Figure 8 demonstrates the pelvic tilt in the frontal plane.

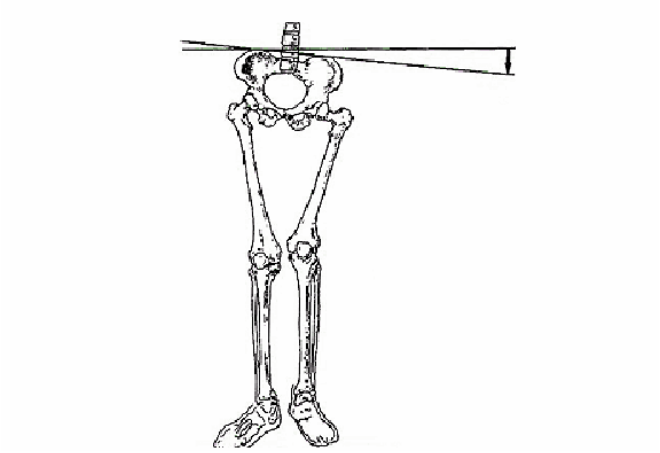


Figure 8. Pelvic Tilt [From Ref. 7.]

Knee flexion is another adjustment of the body that prevents the COG from going even higher.

The foot and ankle motion are perhaps the most important mechanisms in smoothing the trajectory of the COG during the stance phase. They contribute to that task by keeping the trajectory of the COG in a horizontal position as depicted in Figure 9.

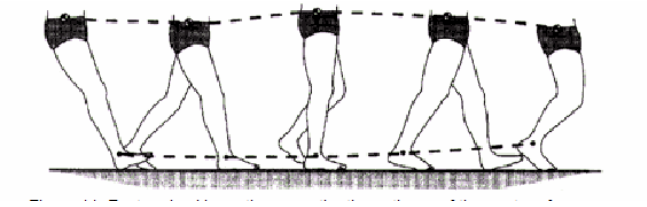


Figure 9. Foot and Ankle Motion [From Ref. 7.]

Knee motion is directly related to foot and ankle motion, and it also helps in

smoothing the trajectory of the COG. In general terms, it could be claimed that, when the ankle is depressed, the knee extends. Similarly, when the ankle is elevated, the knee flexes [Ref. 7], as seen in Figure 10.

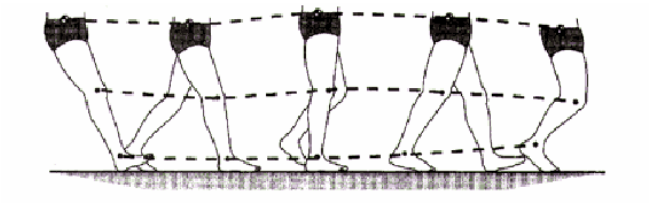


Figure 10. Knee Motion in Synchronization with Foot and Ankle Motion [From Ref. 7.]

Finally, lateral pelvic displacement is the factor that improves the position of the COG over the support limb as depicted in Figure 11.

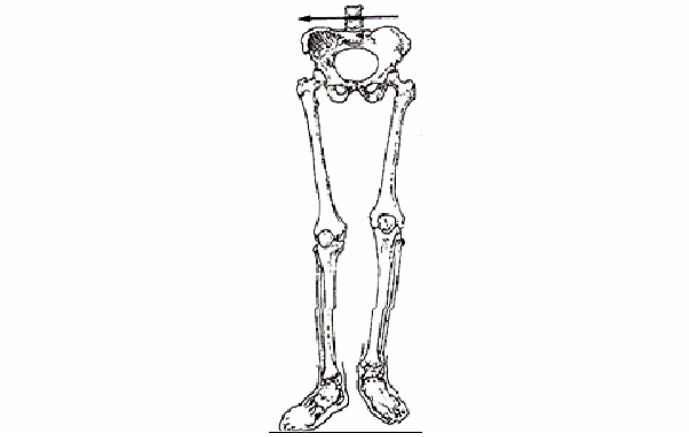


Figure 11. Lateral Pelvic Displacement [From Ref. 7.]

D. SUMMARY

This chapter discussed the basic elements of human walking and provided an insight into human locomotion. The gait–cycle along with its subdivisions was introduced, and additional information concerning the time and distance parameters of walking gait presented. Finally, the last part of the chapter discusses the determinants of gait. The following chapter discusses forward kinematics and, in particular, it introduces manipulator kinematics. Thus, the knowledge obtained from this chapter will be used in the next chapter to derive a mathematical expression of a human walking model representation.

III. MATHEMATICAL MODEL

This chapter provides a brief discussion of kinematics and dynamics. It focuses on the study of forward kinematics, which is used for the derivation of the mathematical model of the human gait. The reason for this is that the kinematic approach is simpler than the dynamic one, and it is more capable of providing a sufficient representation of the locomotion of the lower extremities of a human being. The biomechanics fundamentals presented in the previous chapter are an essential factor in implementing forward kinematics in the human walking model illustrated in this chapter.

The lower part of the human body is modeled as an articulated object constructed of rigid parts (links) connected with each other by joints. In order to estimate the position and the orientation of the rigid parts of the model, a discussion in manipulator kinematics is found to be helpful.

After deriving the mathematical model which computes the position and orientation of the links representing the lower extremities, a Matlab program was written to test the validity of the model based on the numerical solutions. This model was used in the next chapter to simulate human walking based on pseudo–data.

A. BACKGROUND

There are two ways to create physically–based mathematical models for human motion simulation, kinematics and dynamics. Kinematic models are very simple to implement, and do not have as large a computational load as dynamics. Their drawback is that they depend on the geometry of human body parts [Ref. 14]. In dynamic models, all the forces and torques which contribute to the development of the gait are calculated. On the contrary, in kinematic models, the forces and torques are not involved. However, displacements, velocities, and accelerations are studied. Since one of the objectives of this thesis was the study of human walking, it is natural to focus on the kinetics and, especially, on the kinematics of lower limbs (extremities) [Ref. 15].

According to Graig, “Kinematics is the science of motion which treat motion without regard to the forces which cause it” [Ref. 16]. Since the study of kinematics concerns position and all its derivatives, such as velocity and acceleration, kinematics of ma-

nipulators refers to the geometry and the time-based properties of the motion. The human body could be thought of as a mechanical manipulator consisting of many rigid parts connected with joints. Hence, the lower parts of the body could be thought of as a five–rigid–part mechanical manipulator (including a link to describe the pelvis). For this mechanical manipulator, a mathematical model of walking must be derived based on the theory of forward kinematics.

In forward kinematics, given a set of angles, the position and orientation of each part of the mechanical manipulator can be calculated. The calculation of the position and orientation of the last part of the chain, which is called the end–effector, is important. On the other hand, inverse kinematics presents the exact opposite problem. Given the position and orientation of the end–effector, the joint angles are calculated. Generally, the position of the manipulator is described by comparing the axis–frame, which is attached to the end–effector to the axis–frame that is attached to the base of the manipulator [Ref. 16] (Figure 12).

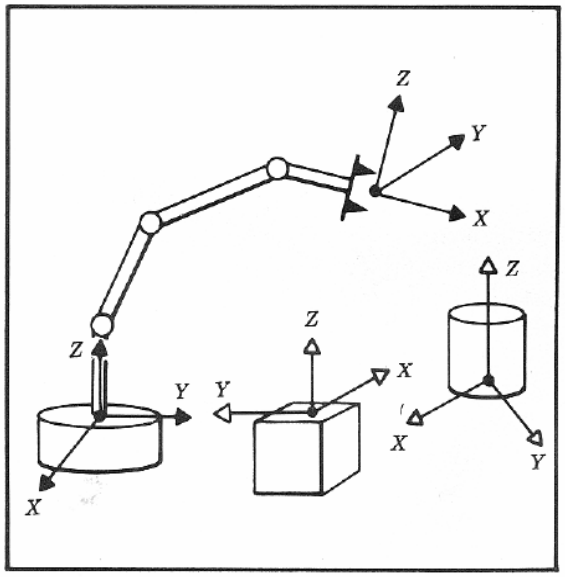


Figure 12. Tool Frame Related to the Base Frame [From Ref. 16.]

Dynamics involves the study of forces and torques in generating the human gait. When the forces and torques are to be determined, given the behavior of each link, the problem is called *the inverse dynamics* problem. On the other hand, when the torques are

given and the accelerations are to be computed, the problem is called *the direct dynamics* problem. Problems such as representing the movements of certain body parts due to the function of the muscles require a dynamics approach such as Newton-Euler mechanics [Ref. 17].

B. MANIPULATOR KINEMATICS

The materials presented in this section are mainly drawn from [Ref. 16].

As previously stated, kinematics deal with the study of motion without involving the forces and the torques applied to the body. It involves the calculation of position as well as the derivatives of position (velocity and acceleration). Kinematics of manipulators refers to the geometry of the body and all of its time–based parameters. The way to handle manipulator kinematics is to define frames on every link and then to study how these frames connect to each other. Another point of study is to see how these frames change as the body articulates.

Manipulators consist of rigid links connected with joints. There are six different types of joints. However, only two of them are used on a grand scale, since they exhibit one degree of freedom, and manipulators are generally constructed with joints that have only one degree of freedom. Therefore, revolute and prismatic joints are the most common (Figure 13). When a joint of a mechanism exhibits n degrees of freedom, it is analyzed into $n-1$ links of zero length connected with joints that exhibit only one degree of freedom. A large number of characteristics of the link exist, including the type of material, strength and shape of the link, and location and type of the joints. Nevertheless, for the purposes of this thesis, a link is considered only as a rigid body which defines the relationship between two neighboring joints.

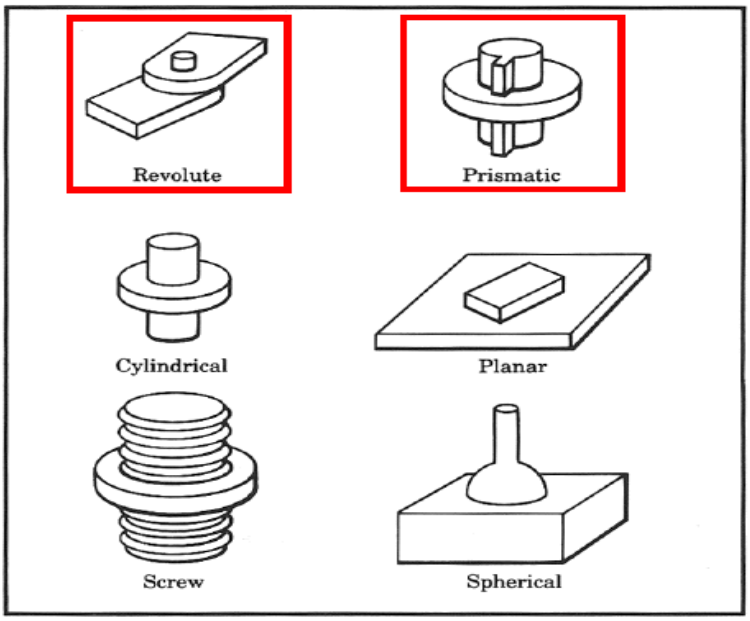


Figure 13. The Six Common Types of Joints [From Ref. 18.]

1. Link Parameters

Before analyzing the link parameters, another definition concerning the joint axis is in order. The joint axis is a line in space about which link *n* rotates with respect to link $n-1$.

In a manipulator, each link has four link parameters: link length, link twist, link offset and joint angle. The length of a line that is mutually perpendicular to both joint axes is called link length. Link length a_{i-1} involves joint axes i and $i-1$.

The second parameter is called link twist. The link length and link twist define the relative location of the two axes. The α_{i-1} angle, measured from axis $i-1$ to axis i, using the right-hand rule, is called link twist.

The following two parameters are used to describe how two neighbor links are connected to each other. Neighbor links have a common joint axis in the middle. The first parameter refers to the distance measured along the common axis between two neighbor links (axis i), from the point where a_{i-1} intersects axis i to the point where a_i intersects axis i, and is called the link offset. The link offset at link axis i is d_i.

The second parameter refers to the relative angle between one link and its neighbor and is called the joint angle θ_i. Out of four link variables, the first two describe the link itself, and the other two describe how this link is connected to its neighbor links. Figure 14 illustrates the link parameters of one link of a manipulator.

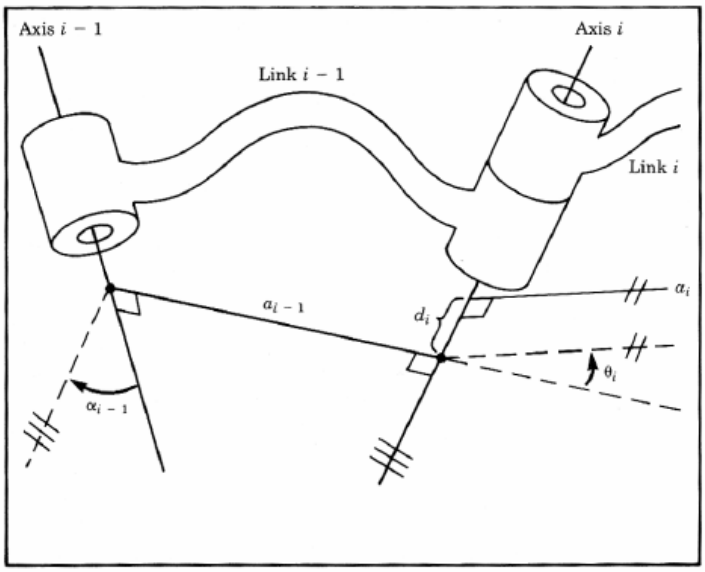


Figure 14. Link Parameters [From Ref. 18.]

2. Convention in Locating Frames

One frame is attached to each link. In order to describe the relations between neighboring links, the following convention is followed.

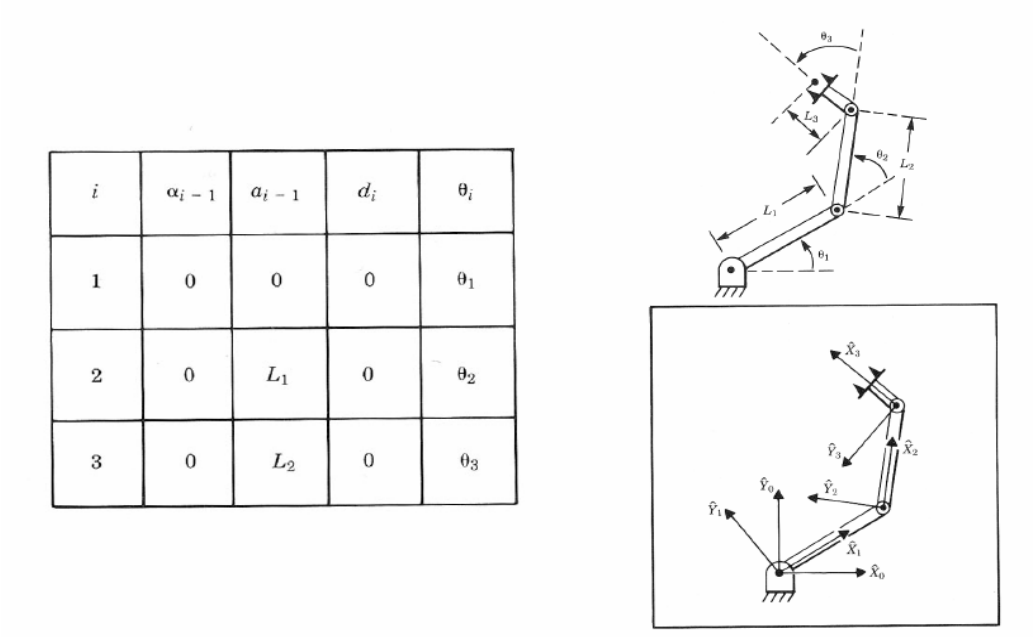
Assume that the i-th frame is attached to the i-th link. It is a three–dimensional frame where the *Z*–axis is the same as the joint axis i. The *X*–axis is pointing towards the direction of a_i, where a_i is the length of link i. The direction of the *Y*–axis is determined based on the right–hand rule. Note that the origin of the frame coincides with the point where a_i intersects the joint $i-$axis.

An alternative definition of the link parameters according to the frame convention is given below.

- a_i refers to the distance from Z_i to Z_{i+1} measured along X_i.
- α_i refers to the angle between Z_i and Z_{i+1} measured along X_i.

- d_i refers to the distance from X_{i-1} to X_i measured along Z_i.
- θ_i refers to the angle between X_{i-1} and X_i measured along Z_i.

Figure 15 illustrates a three–link manipulator with all its joints being revolute. There are three frames attached, one for each limb, plus the reference zero–frame for a total of four frames. In addition, every link length and joint angle is depicted along with the corresponding link parameters.



i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3

Figure 15. Three-Link Manipulator [From Ref. 18.]

3. Mappings Involving Translated and Rotated Frames

Before presenting the mathematical model of the manipulator, which represents the orientation and position of the lower extremities, a small digression is in order. It is important first to gain insight into the mappings involving translated and rotated frames, and second to understand the meaning and the utility of transformation matrices.

When pure translation occurs, a point P in space, relative to an initial frame {A}, is described as follows[2].

$$^{A}P = {}^{B}P + {}^{A}P_{BORG} \tag{3-1}$$

[2] Notation is such that vectors have leading superscripts and matrices have leading superscripts and subscripts.

where ${}^{A}P$ is the vector that locates point P relative to frame {A}, ${}^{B}P$ is the vector that locates point P relative to frame {B}, and ${}^{A}P_{BORG}$ is the vector that locates the origin of frame {B} relative to frame {A}.

When a pure rotation occurs, a point P in space, relative to an initial frame {A}, is described as follows.

$$ {}^{A}P = {}^{A}_{B}R\,{}^{B}P \tag{3-2} $$

where ${}^{A}_{B}R$ is the rotation matrix that describes frame {B} relative to frame {A}. However, there are cases in which both translation and rotation occur (Figure 16). Therefore, the mapping of a vector from its description in one frame to a description to another frame is given as.

$$ {}^{A}P = {}^{A}_{B}R\,{}^{B}P + {}^{A}P_{BORG} \tag{3-3} $$

or

$$ {}^{A}P = {}^{A}_{B}T\,{}^{B}P \tag{3-4} $$

where

$$ {}^{A}_{B}T = \begin{bmatrix} {}^{A}_{B}R & {}^{A}P_{BORG} \\ 0\ 0\ 0 & 1 \end{bmatrix}. \tag{3-5} $$

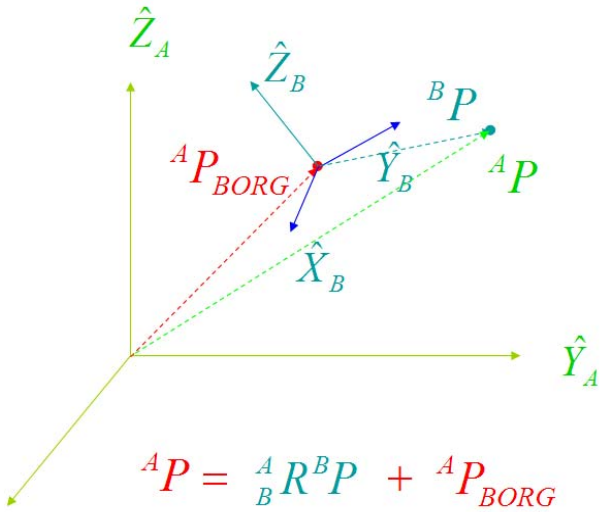


Figure 16. General Transformation [After Ref. 19.]

This matrix includes the rotation and translation of the general case. Thus, transformation matrices are used to specify a frame. In the case of manipulator kinematics, the transformation matrix, which relates two frames attached to neighbor links, is given by

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i\sin\alpha_{i-1} \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i\cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3-6} $$

Therefore, the first step is to define the frames attached to each link. Next, the corresponding link parameters to each frame must be calculated. Then, the transformation matrices corresponding to each set of link parameters can be derived. Finally, by multiplying the individual transformation matrices from ${}^{0}_{1}T$ to ${}^{N-1}_{N}T$, the general transformation matrix ${}^{0}_{N}T$ can be obtained as

$$ {}^{0}_{N}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T \ldots\ldots {}^{N-2}_{N-1}T\,{}^{N-1}_{N}T\,. \tag{3-7} $$

C. MATHEMATICAL MODEL FOR HUMAN LOWER EXTREMITIES

Figure 17 illustrates a manipulator model which describes the lower extremities of the human body and consists of five links. Each leg consists of two links, one thigh and one shank. One link is assigned to represent the pelvis. The foot is not included in this

model due to the complexity and the significant computational load it would cause. Furthermore, after deriving the mathematical model, a computer simulation of human walking is illustrated, using pseudo–data, in order to calculate the traveled distance. Next, real data is used by utilizing the MARG sensors.

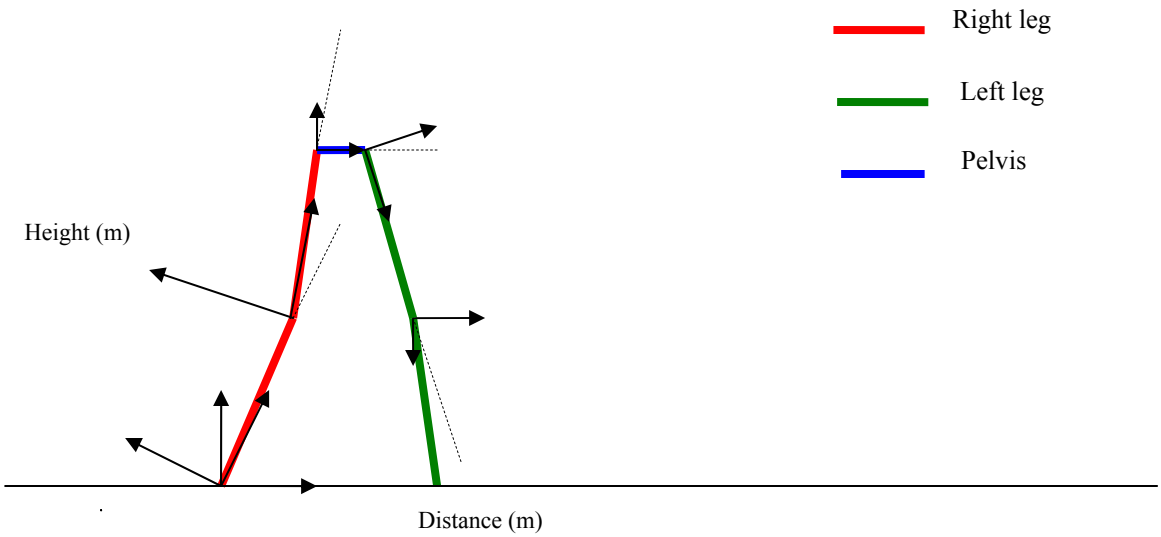


Figure 17. Manipulator Describing Lower Extremities

Figure 17 also shows a two–dimensional scheme in which the lower extremities are in the midstance, right before the double–support phase. Link one represents the right shank, link two represents the right thigh, link three represents the pelvis, link four represents the left thigh and finally, link five represents the left shank. The links are connected to each other with revolute joints. Link one is connected with a revolute joint to the base of the manipulator (ground).

As is shown, the frames are attached to the links. Also the *X*–axes coincide with the links, and the *Y*–axes form a 90–degree angle with the *X*–axes and *Z*–axes being vertical to the plane indicating that all link twists are equal to zero.

Table 2 presents the link parameters of each frame, where l_{pel} the length of the pelvis and φ is the angular displacement of the pelvis in the *Y*–axis.

	a_{i-1}	α_{i-1}	d_i	θ_i
Link 1	0	0	0	θ_1
Link 2	0	l_1	0	θ_2
Link 3	0	l_2	0	θ_3
Link 4	0	$l_{pel}\sin\varphi$	0	θ_4
Link 5	0	l_3	0	θ_5

Table 2. Link Parameters

Next, the individual transformation matrices for each pair of neighboring links are defined. For frames zero and one, the transformation matrix ${}^{0}_{1}T$ is as follows:

$$ {}^{0}_{1}\boldsymbol{T} = \begin{bmatrix} \cos\boldsymbol{\theta}_1 & -\sin\boldsymbol{\theta}_1 & 0 & 0 \\ \sin\boldsymbol{\theta}_1 & \cos\boldsymbol{\theta}_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-8} $$

and, for frames one to two,

$$ {}^{1}_{2}T = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & l_1 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3-9} $$

Before defining the transformation matrix ${}^{2}_{3}T$, a brief discussion concerning the motion of the pelvis is in order. Let l_{pel} be the length of the pelvis. During a step forward, the maximum angular displacement in the *X*–axis is approximately three degrees. The corresponding angular displacement in the *Y*–axis is minus three degrees. As previously

stated, this is a two–dimensional approach. Therefore, frame three can be defined as shown in Figure 17, without any loss of generality. From the values of θ_1, θ_2 and φ, the angle θ_3 is calculated by using:

$$\theta_3 = 360 - \theta_2 - \theta_1 + \varphi . \tag{3-10}$$

Hence, the transformation matrix ${}^{2}_{3}T$ is equal to:

$${}^{2}_{3}T = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & l_2 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3-11}$$

Similarly,

$${}^{3}_{4}T = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & l_{pel}\sin\varphi \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-12}$$

and

$${}^{4}_{5}T = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & l_3 \\ \sin\theta_5 & \cos\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3-13}$$

Finally, Equation (3-8) gives the transformation matrix that relates frame five to frame zero:

$${}^{0}_{5}T = \begin{bmatrix} \cos(\theta_1+\ldots+\theta_5) & -\sin(\theta_1+\ldots+\theta_5) & 0 & \begin{array}{c} l_3\cos(\theta_1+\ldots+\theta_4)+l_{pel}\sin\varphi\cos(\theta_1+\ldots+\theta_3) \\ +l_2\cos(\theta_1+\theta_2)+l_1\cos\theta_1 \end{array} \\ \sin(\theta_1+\ldots+\theta_5) & \cos(\theta_1+\ldots+\theta_5) & 0 & \begin{array}{c} l_3\sin(\theta_1+\ldots+\theta_4)+l_{pel}\sin\varphi\sin(\theta_1+\ldots+\theta_3) \\ +l_2\sin(\theta_1+\theta_2)+l_1\sin\theta_1 \end{array} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3-14}$$

The coordinates of the end–effector with respect to frame five are:

$$^{5}P = [l_4 \quad 0 \quad 0 \quad 1]^T . \tag{3-15}$$

Hence, by using Equation (3-5), the coordinates of the end–effector with respect to the reference frame zero are:

$$^{0}P = \begin{bmatrix} l_4\cos(\theta_1+...+\theta_5)+l_3\cos(\theta_1+...+\theta_4)+l_{pel}\sin\varphi\cos(\theta_1+...+\theta_3) \\ \quad +l_2\cos(\theta_1+\theta_2)+l_1\cos\theta_1 \\ l_4\sin(\theta_1+...+\theta_5)+l_3\sin(\theta_1+...+\theta_4)+l_{pel}\sin\varphi\sin(\theta_1+...+\theta_3) \\ \quad +l_2\sin(\theta_1+\theta_2)+l_1\sin\theta_1 \\ 0 \\ 1 \end{bmatrix} . \tag{3-16}$$

1. Testing the Model

In order to verify that the mathematical model presented above is correct, a program was written in Matlab. This code evaluated the model by checking the results of the numerical solutions. The code is presented in Appendix A.

Table 3 illustrates the fixed link parameters used in the code.

	a_{i-1} (degrees)	α_{i-1} (meters)	d_i (meters)	θ_i (degrees)
Link 1	0	0	0	60
Link 2	0	0.45	0	30
Link 3	0	0.4	0	270
Link 4	0	$0.2\sin\frac{\pi}{60}$	0	300
Link 5	0	0.4	0	345

Table 3. Fixed Link Parameters

After running the program, the following results are obtained.

$$ {}^{0}_{2}P = \begin{bmatrix} 0.2250 \\ 0.7897 \\ 0.0000 \\ 1.0000 \end{bmatrix}, \quad {}^{0}_{3}P = \begin{bmatrix} 0.2355 \\ 0.7792 \\ 0.0000 \\ 1.0000 \end{bmatrix}, \quad {}^{0}_{4}P = \begin{bmatrix} 0.4355 \\ 0.4433 \\ 0.0000 \\ 1.0000 \end{bmatrix}, \text{ and } {}^{0}_{5}P = \begin{bmatrix} 0.5519 \\ 0.0086 \\ 0.0000 \\ 1.0000 \end{bmatrix}. \qquad (3\text{-}17) $$

Each column matrix is a set of coordinates for the corresponding end–effector with respect to the reference frame zero. The fourth element of all vectors is equal to one. This results from Equation (3-5) where a "one" is added in order to make ${}^{0}_{N}P$ compatible to the 4×4 transformation matrix. The third element of all vectors is equal to zero. This is absolutely logical since there is no motion in the *Z*–dimension. The first and the second elements give the coordinates of the corresponding end–effector in the plane. If those coordinates are compared to Figure 17, it is possible to state that the results are fairly reasonable and, therefore, the model works quite satisfactorily. For example, it appears that the end–effector is slightly above the ground with $y = 0.0086$ meters.

D. SUMMARY

This chapter provided a brief description of forward kinematics theory, discussed robotics fundamentals and, especially, provided some insight into manipulator kinematics. In addition, a mathematical model for a human manipulator was developed. The human manipulator described the geometry of the lower limbs and the pelvis. The mathematical model was used to define the position and orientation of the end–effector. Finally, a Matlab program was written to demonstrate the correctness of the model.

The next chapter consists of two parts. The first part presents the approaches taken so far in order to simulate human walking and to estimate the position of a person. The second part presents a computer simulation of human walking. Matlab code implements the mathematical model presented in this chapter, which uses pseudo–generated data to simulate a full gait–cycle and to calculate the distance walked along the horizontal axis.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. A FOUR–SEGMENT COMPUTER SIMULATION OF NORMAL HUMAN GAIT

This chapter outlines the model that represents the lower parts of the human body and that was used in a computer simulation of the normal human gait. This model is slightly different from the mathematical model derived in the previous chapter. It is more restrictive so that the movement can occur only in the sagittal plane. At the beginning of the chapter, several other approaches of computer simulations of human walking are presented. The main part of this chapter discusses the methodology as well as the results of the simulation.

A. APPROACHES OF HUMAN GAIT SIMULATION

Two major subdivisions exist in order to study the human gait: kinetics and kinematics. This chapter focuses on the study of kinematics involved in human walking. For any simulation to be successful, a reasonable kinematic pattern should be provided. For the human gait, a simulation should include both single and double support phases [Ref. 20].

In the past, several simulations have taken place for many purposes. Some were to achieve clinical–surgical goals [Ref. 21], while some others have different objectives [Ref. 22].

Another issue is the interest that some investigators show for a specific phase of the gait cycle, or even a specific subdivision of a phase. Zarrugh and Radcliffe [Ref. 21] have focused on the simulation of the swing phase, while Henami, Zheng, and Hines [Ref. 23] deal with the initial position of the body and, in particular, of the lower limbs during the initiation of walking. According to Gilchrist and Winter [Ref. 20], dealing with only one phase instead of the entire gait cycle is more convenient in terms of the numerical load that occurs when the foot goes from the swing phase to the stance phase or, in other words, from no loading to full loading.

In some other cases, the model which represents the lower extremities is simplified. This translates to either the representation of the leg with one rigid bar instead of two (shank and thigh) or a model with no feet. It is very common for investigators not to

include the foot segment in the model. Specifically, Ju and Mansour [Ref. 24] introduced a double limb model for the leg, without the foot, in order to simulate the support phase of the human gait. Furthermore, Amirouche, Ider, and Trimble [Ref. 25] followed the same model in their attempt to simulate and analyze human locomotion. In other experiments, the supporting extremity consists of two rigid bars which represent the thigh and the shank. The foot is not included while the lower part of the shank is fastened to the ground [Refs. 26 and 27].

Another major element in these kinds of experiments is whether the model is two–dimensional or three–dimensional. Individually, each limb undergoes a planar movement. However, both legs are connected to each other with the pelvis. Since the pelvis is perpendicular to the legs, inevitably, the overall movement becomes three–dimensional. This happens because the pelvis moves in both the frontal and the horizontal plane. Some researchers prefer to disregard the pelvis in order to simplify their model, which leaves a two–dimensional model [Ref. 28]. Apart from the importance of the pelvis as far as the motion of the lower part of the body is concerned, it is also important because it minimizes the amount of motion of the HAT (head, arms, and trunk) during the gait cycle. It is obvious that the results are better and far more accurate in the case where the pelvis is included but, at the same time, the computational load increases.

A very common tactic is for the investigators to use predetermined trajectories of one or more segments as feedback [Ref. 20]. The feedback may be a part of a function that optimizes the results. Davy and Audu use this technique to predict muscle forces during the swing phase [Ref. 29].

Meglan presented a simulation of human walking that is very close to the ideal one. The model consisted of segments representing the entire human body, and not only the lower part. It is also a three–dimensional model and it includes the feet [Ref. 30].

B. INTRODUCTION OF THE FOUR–SEGMENT MODEL

One of the main objectives of this thesis was to simulate human walking. This was accomplished by creating a program in Matlab and by implementing a model which represents the human body. The previous section of this chapter presented several ap-

proaches concerning the simulation model. All of them had advantages and drawbacks while, in many cases, the investigator had to consider the tradeoffs that appeared.

In this project, a computer simulation of the human walking was developed. For this purpose, the kinematic approach was adopted. The kinematic approach is much simpler than the dynamic approach, and human movement can be sufficiently represented using this kind of approach. According to Cheng and Moura [Ref. 31], there are two basic elements that have to be considered before creating a model for human walking. First, there is the model of the human body which gives the geometrical aspect of the problem. Second, there is the model of the walking, which provides the topological aspect of the problem. These two components are used in order to synthesize the walker.

The human body can be represented as a multi–link object connected by joints and rigid bars. Models of this kind vary from 10 to 17 links. Each link is connected to its neighbor with a joint. The more articulated the parts and degrees of freedom, the more accurately the motion of the human body is described. On the other hand, this increases the number of the required parameters. Thus, it increases the complexity of the estimation of human movement [Ref. 31]. The model used in this project is a model stick figure of the lower limbs, and it represents only the lower part of the body. The model shown in Figure 18 consists of four joints and joint angles θ_i (where $i = 1,..,4$). One link is for the right shank, one for the right thigh, one for the left shank, and one to represent the left thigh.

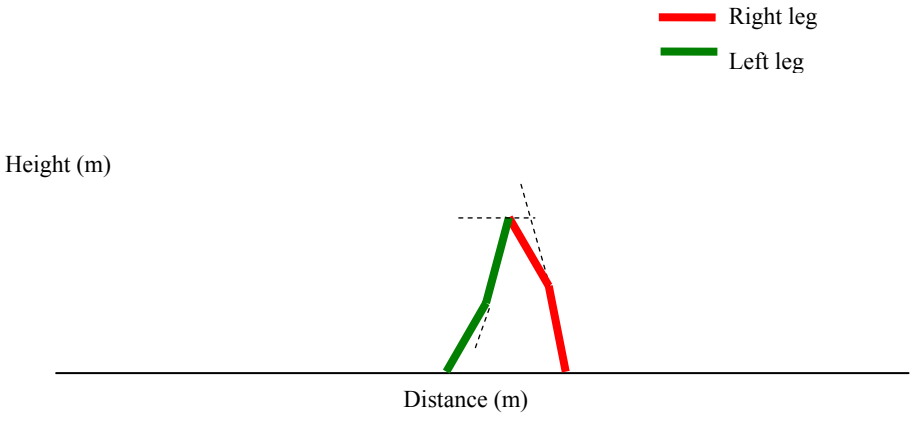


Figure 18. Simulation Model

As shown in Figure 18, the model is two–dimensional and the motion of the lower body occurs only in the sagittal plane. In addition, it is clear that there is no segment to represent the pelvis, which is chosen to keep the model in two dimensions. Otherwise, due to the limited but significant movement of the pelvis in both the frontal and horizontal plane, the overall motion would be in three dimensions. This would increase the computational load and the number of parameters required to describe the motion. Moreover, the model depicted in Figure 18 does not include any segments to represent the feet since the model used for the simulation is nothing else but a simplified version of the mathematical model derived in the previous chapter. That model did not include any segments to represent the feet either. Hence, it is possible to avoid complex calculations and difficult derivations.

A brief discussion concerning the initial position of the model is in order. Figure 19 illustrates the initial position of the model.

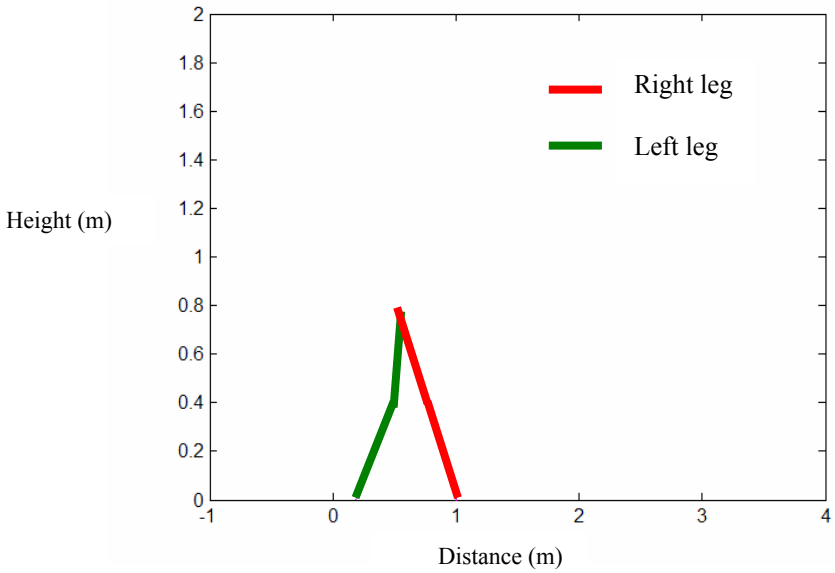


Figure 19. Initial Position of the Model

In the initial position, the model is at the beginning of the stance phase, right where the first double support of the gait–cycle occurs. Assuming that the reference leg is the right leg, it is fastened to the ground. If there were feet included in the model, then it could be claimed that this initial position is described best as a heel strike, which is the instant that the leading extremity strikes the ground. The traditional terminology of the

gait–cycle introduced in Chapter II, will be applied in this chapter, in order to better describe the results of the walking simulation. The lengths of the links are constant as illustrated in Table 4. The supporting extremity touches the ground at (1,0) while, according to the theory of kinematics, the coordinates of the joint as well as the coordinate of the end–effector are determined by the joint angles. Table 4 also presents the initial set of angles.

Body Parts	**Symbols**	**Length in m**	**Joint angles**	**Value in degrees**
Right Shank	l_1	0.45	θ_1	$-\frac{\pi}{6}$
Right Thigh	l_2	0.4	θ_2	$-\frac{\pi}{6}$
Left Shank	l_3	0.4	θ_3	$-\frac{11\pi}{12}$
Left Thigh	l_4	0.45	θ_4	Determined by θ_1, θ_2, θ_3

Table 4. Lengths of Links and Values of Initial Angles

C. RESULTS OF THE SIMULATION

After running the program in Matlab, a window appears and the model executes the amount of gait–cycles according to the code. (This program is illustrated in Appendix B.) The following five figures illustrate some characteristic frames of the movement of the model. As previously mentioned, the coordinates of the model change with respect to the angles. Therefore, since no real data are available yet, pseudo–data must be introduced to move the model. This pseudo–data is based on the theory of human gait–cycle, which was developed in the previous chapter. Their duration and their subdivisions and the generation of pseudo–data is achieved from the theory of biomechanics provided in Chapter II regarding the phases of the gait–cycle. Next, the gait–cycle is divided into n frames from which approximately $0.6n$ frames are used to describe the stance phase, and $0.4n$ frames are used to describe the swing phase.

In Figure 20, the model is at stance phase and, more specifically, it is in foot flat which occurs right after the heel strike. By noticing that the left leg has already traveled a distance, it is possible to claim that the frame of Figure 20 depicts a moment between the foot flat and the midstance.

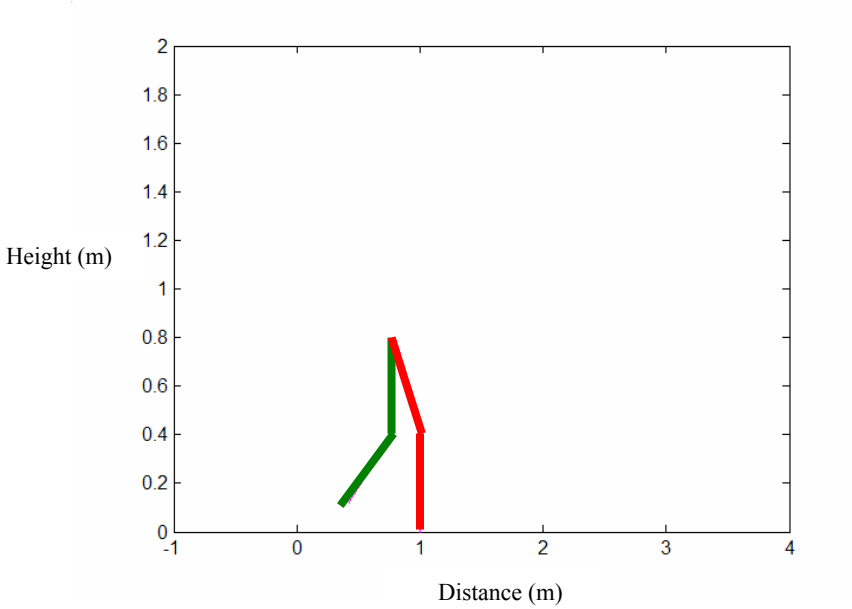


Figure 20. Foot Flat

In Figure 21, the model is at midstance at which the body weight is directly under the supporting lower extremity [Ref. 5]. The reference leg is still fastened to the ground.

In Figure 22, the reference extremity is still fastened to the ground but if there was a foot attached, this would be the instant where the heel of the reference extremity leaves the ground or heel off. A comparison between Figure 22 and the other figures describing the heel off, shows that this particular frame describes the model as being in the middle of the duration of the heel strike, just prior to the extension of the left leg.

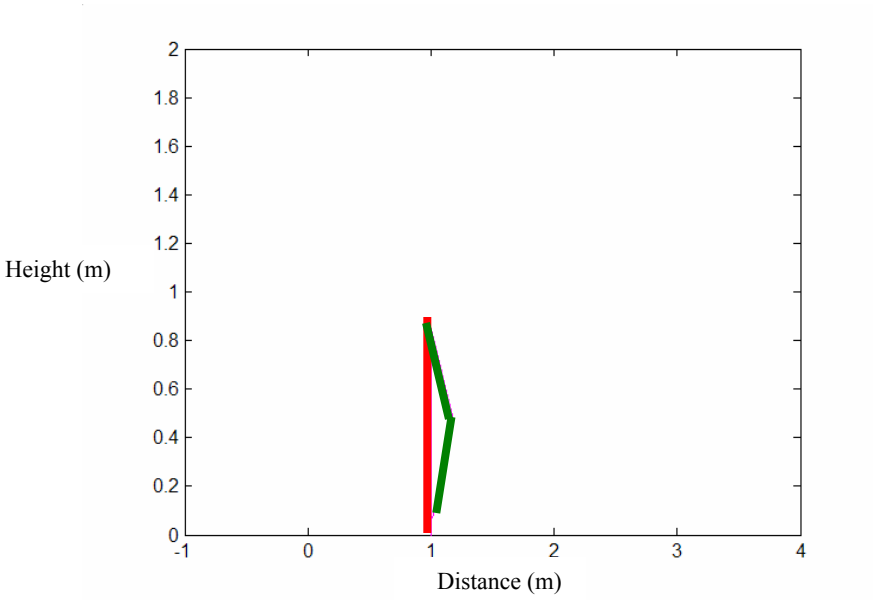


Figure 21. Midstance

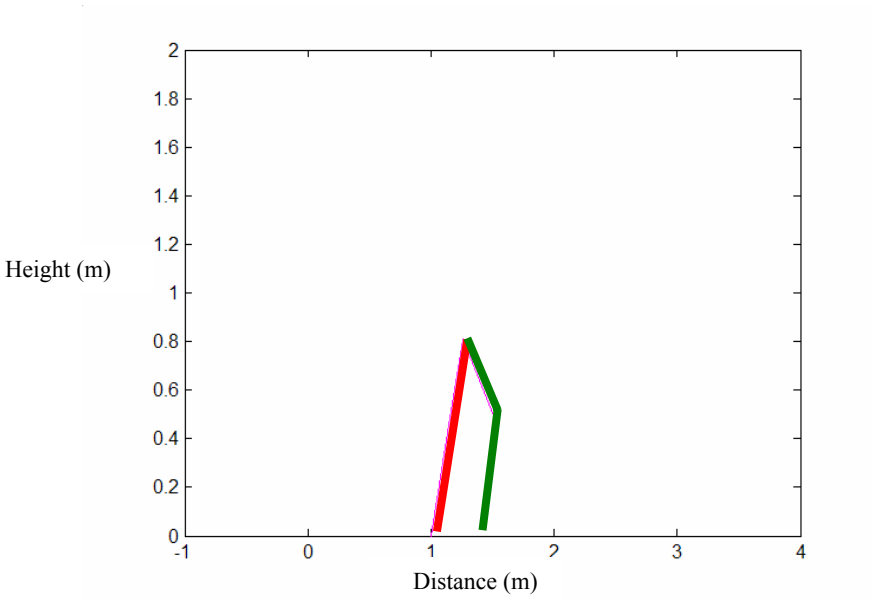


Figure 22. Heel Off

Figure 23 illustrates the instant when both the lower extremities touch the ground again (double support). This is the last subdivision of the stance phase called toe off. In toe off, only the toe of the reference extremity touches the ground. From that point on, the left leg will be fastened to the ground.

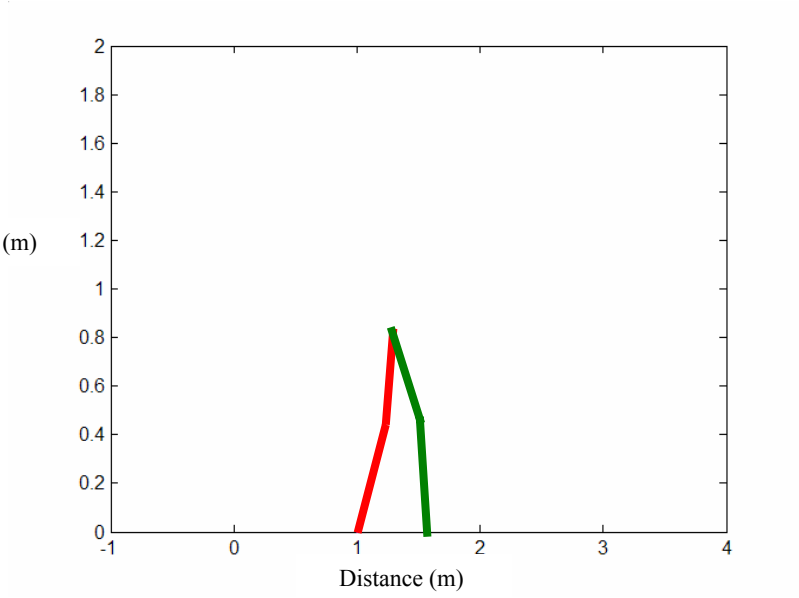


Figure 23. Toe Off

In Figure 24, the model has entered the swing phase. In particular, it is in acceleration and, to be more specific, the figure illustrates the swinging extremity as directly under the body.

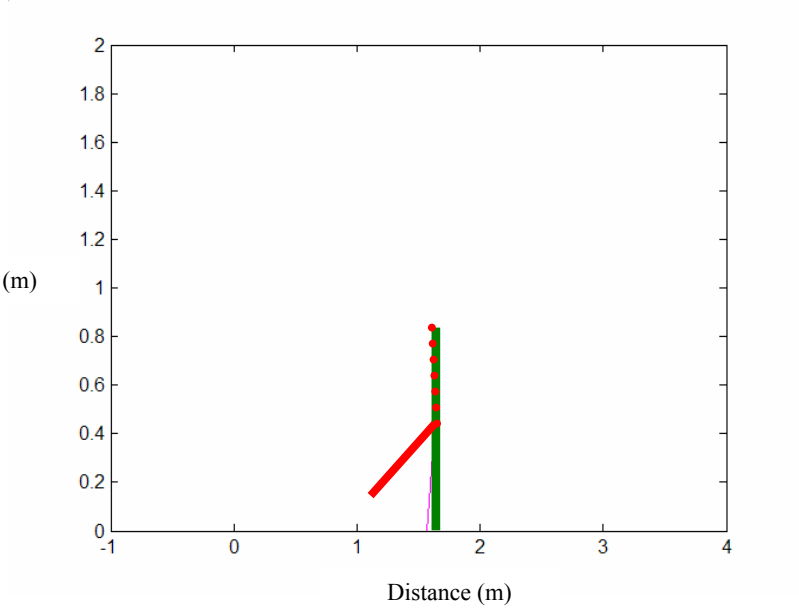


Figure 24. Acceleration

Figure 25 illustrates the model in the midswing. This is the moment when the swinging extremity passes directly beneath the body.

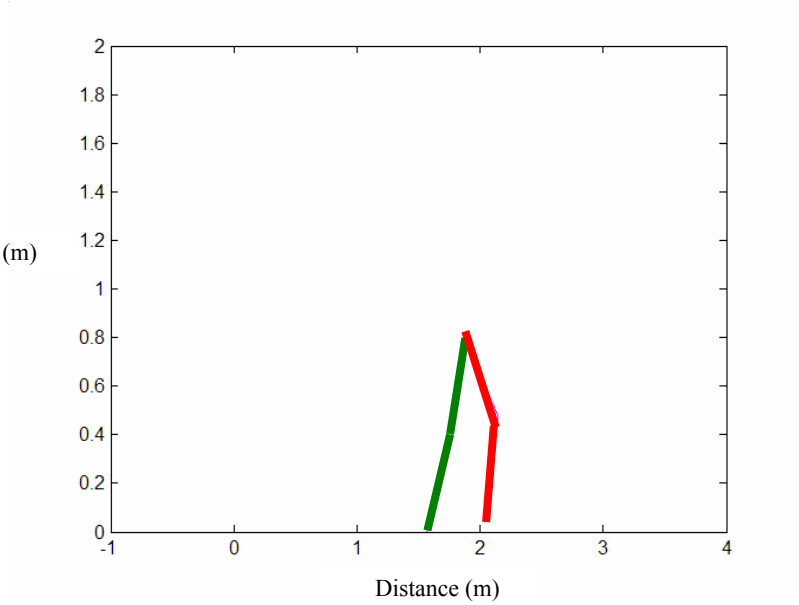


Figure 25. Midswing

Figure 26 illustrates the final frame of the first gait–cycle of the model. It is called deceleration, and occurs immediately after midswing. Note that the knee is extending and it is preparing for another heel strike, which will conclude the first gait–cycle and initiate the second.

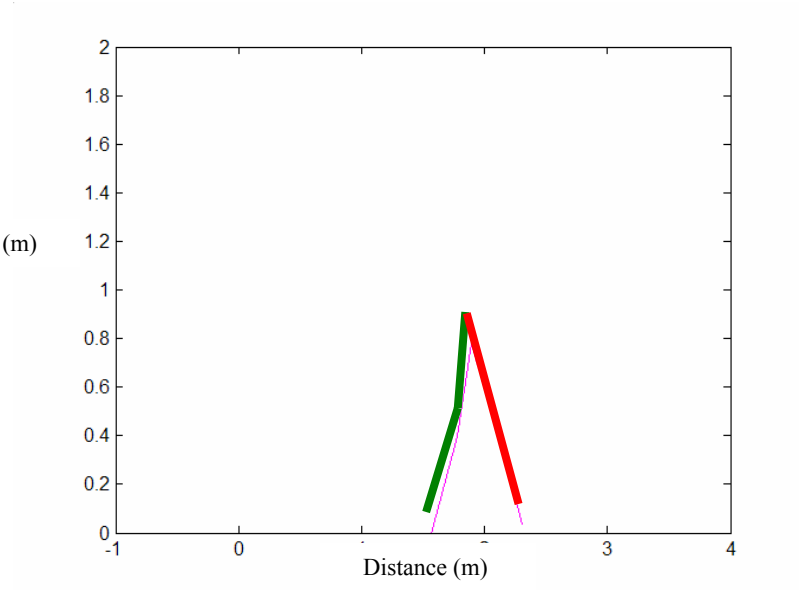


Figure 26. Deceleration

Figures 27 and 28 illustrate the decelerations for the second and the fourth gait–cycle, respectively.

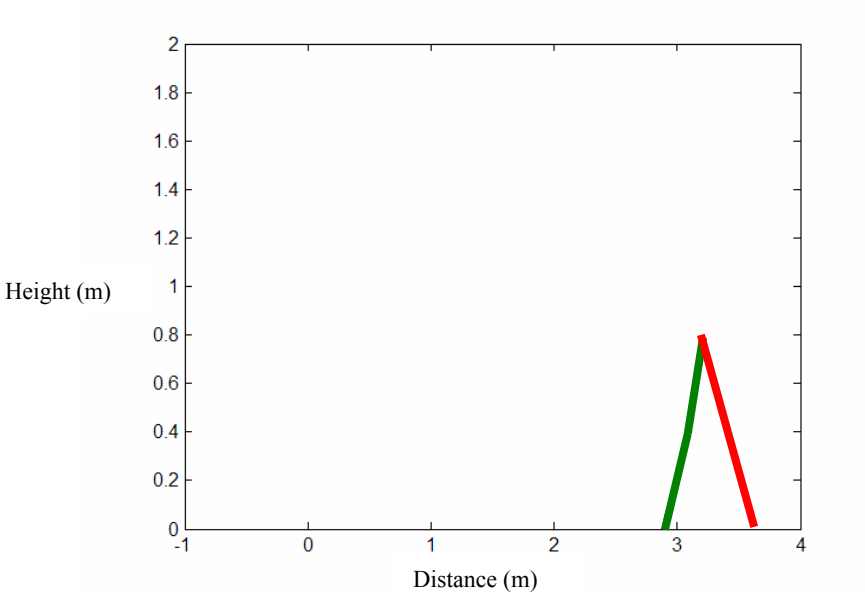


Figure 27. Deceleration of 2nd Gait–Cycle

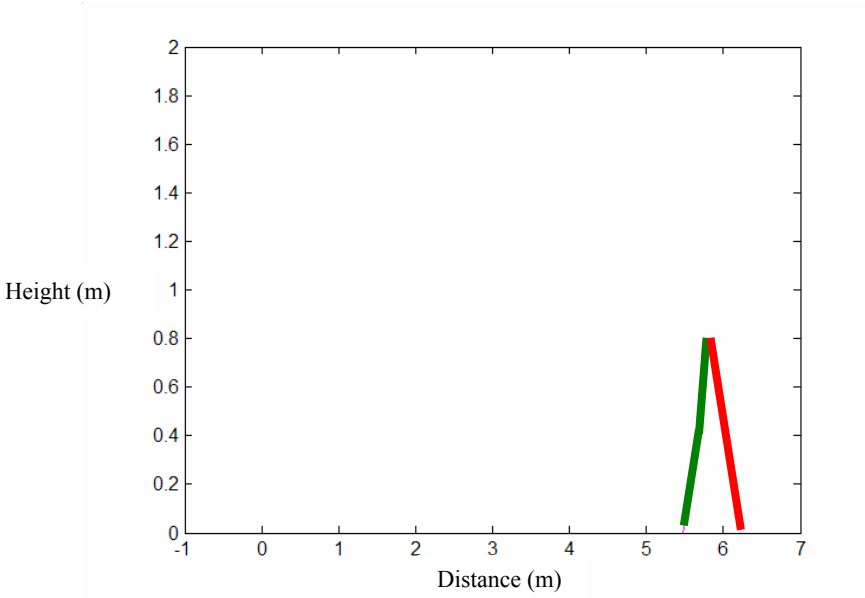


Figure 28. Deceleration of the 4th Gait–Cycle

Another attribute of the code is that it can calculate the total distance traveled. Table 5 illustrates the distances traveled for specific numbers of gait–cycles.

Number of Gait–Cycles	Total Distance Traveled [m]
1	1.3049
2	2.6098
3	3.9147
4	5.2196
6	7.8294
12	15.6588

Table 5. Traveled Distances

In addition, the Matlab code can be modified to "command" the model to move with a changeable step. However, the possibilities of a wide variety of step lengths and flexibility in the motion patterns are rather limited due to the strict rules of biomechanics implemented for the derivation of the pseudo–data.

D. SUMMARY

Several approaches made by different investigators in simulating human motion, and, in particular, human walking were discussed in the beginning of this chapter. The mathematical model derived in the previous chapter was simplified by subtracting the segment that represented the pelvis in order to further decrease the computational load as well as to limit the motion of the model in the sagittal plane only. The simplified model was then implemented in a code written in Matlab. In addition, pseudo–data was generated to represent the joint angles, which were a crucial parameter of the model since a kinematics approach was chosen from the beginning. This chapter illustrated several frames of the gait–cycle of the simulation. The model executed as many gait–cycles as the code demanded. Moreover, the total distance traveled at the end of every gait cycle was calculated. Slight changes in the code were made so that the model could achieve a changeable step.

In the next chapter a more sophisticated model is illustrated. It consists of two shanks, two thighs, two feet and a pelvis. The feet are mass rigid segments and modeled as triangles. In addition a program in LISP is presented providing the movement of a foot also modeled as a triangle. Finally a comparison between the outcomes of those two different approaches takes place.

V. ENHANCED MATLAB MODEL AND COMPARISON WITH A LISP MODEL

In the previous chapter, a computer simulation of the normal human gait was introduced. The simulation implemented a mathematical model into a Matlab program, which generated pseudo–data to represent the joint angles. Many figures representing several subdivisions of the gait–cycle were illustrated. In addition, another function of the Matlab program made the precise calculation of the total distance traveled possible.

This chapter presents a more advanced and sophisticated simulation of the human gait–cycle. The model consists of seven segments instead of five. Thus, this model also contains two feet and a pelvis. The main concept of the simulation remains the same. Pseudo–data of angles, generated from a program in Matlab based on the same principles of biomechanics, causes the movement of the augmented model.

Next, a simulation of the human gait–cycle is presented in LISP. The model used consists of only one foot. A program written in LISP causes the foot to move, and to show its positions through the various phases of the gait–cycle. Finally, a comparison between the two approaches, Matlab and LISP, is provided.

A. DESCRIPTION OF AN ENHANCED MODEL IN A MATLAB SIMULATION

The simulation presented in the previous chapter managed to describe the phases and the subdivisions of the human gait–cycle successfully. However, it is certain that the kinematic pattern would greatly improve with the addition of a foot. As the number of segments that represents the lower extremities increases, so does the number of degrees of freedom and, consequently, the accuracy of the description of the motion of the human body. It is obvious that in a simulation where the feet are not included, and the lower part (shank) of the reference extremity is fastened to the ground, the results do not have the accuracy of a simulation that implements a model which contains the feet. The most significant problem that arises regarding the addition of the feet is the increase of the required parameters and, therefore, the increase of the calculations required to describe the movement of the body.

In addition to the feet, the problem becomes even more complicated when a pelvis is added to the already existing model. Two issues must be addressed when a pelvis is added. The first issue is the inevitable increase of the computational load, and the second is the manner in which the movement of the pelvis will be simulated, given that the movement occurs in the sagittal plane. Apparently, the addition of the pelvis increases the accuracy in describing the movement of the body dramatically. Furthermore, the pelvis performs a number of significant functions, such as supporting the HAT (head, arms, and trunk), lateral pelvic tilt, and pelvic rotation [Ref. 5]. Despite the addition of the pelvis, the movement of the model is still described in two dimensions and occurs in the sagittal plane. As mentioned in Chapter III, during the movement of the human body, angular displacements of the pelvis occur in both the *X*–axis and *Y*–axis, and reach their maximum values (± 3 degrees) at the double support phase. Therefore, in order for the movement of the pelvis to be accurately described in the sagittal plane, the projections of the pelvis of the *Z*–axis to the *X* and *Y* axes are calculated. Thus, the segment representing the pelvis is not constant. It is changeable, and its size and angles vary.

The lower extremities (including the pelvis) are also represented as a multi–link object, connected with joints and rigid bars. The augmented model used in this chapter is also a model stick figure of the lower limbs. The difference is that the feet are not represented by sticks but by triangles, in order to describe the movement of the feet and their angular relationship to the ankles better. In addition, the description of the feet as triangles results in a more realistic simulation, since they are considered as mass rigid segments. Therefore, the augmented model consists of five links and two triangles. Two links represent the shanks, two links represent the thighs, one link with changeable length represents the pelvis and, finally, two triangles represent the mass rigid feet. Figure 29 illustrates the initial position of the augmented model.

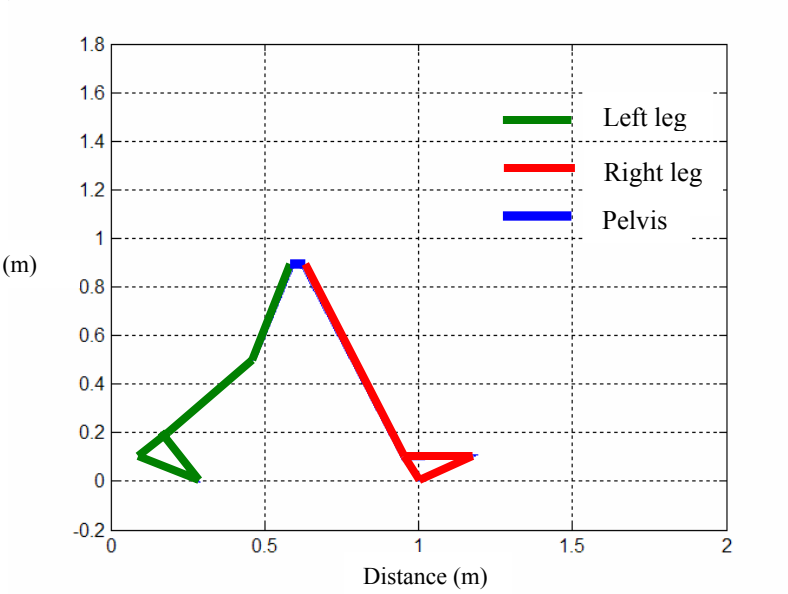


Figure 29. Initial Position of the Augmented Model

As shown in Figure 29, the augmented model is still two dimensional with the difference that the pelvis is represented by one link of changeable length. At this point of double support, the length of the projection of the pelvis reaches its maximum value. The initial position of the augmented model is at the beginning of the stance phase. Assuming that the reference leg is the right one (red–colored), the initial position occurs right when the leading extremity strikes the ground (heel strike).

B. RESULTS OF THE SIMULATION

Following the exact same procedure as in the previous chapter, the augmented model executes the movement according to the Matlab program. (The program for the simulation presented in this chapter is illustrated in Appendix C.) The following figures illustrate some characteristic frames of the movement of the model.

In Figure 30, the model is in foot flat. In the previous chapter, this could only be estimated, since there was no segment to represent the foot. However, in this case, the figure verifies the given description since the foot of the supporting extremity is in full contact to the ground. Furthermore, the length of the projection of the pelvis is decreased.

In Figure 31, the augmented model is in midstance and, as shown in the figure, the pelvis is almost vertical to the sagittal plane.

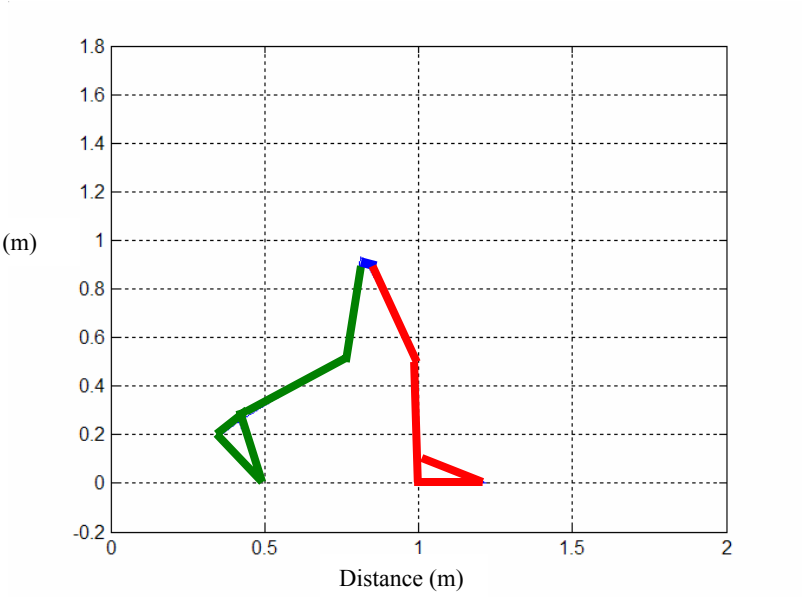


Figure 30. Foot Flat

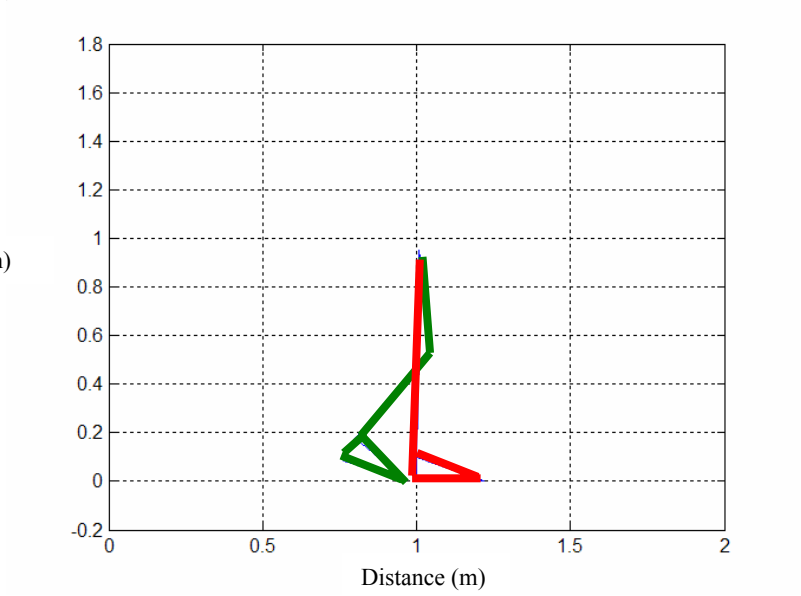


Figure 31. Midstance

In Figure 32, the heel of the reference extremity leaves the ground. This is called *heel off*. In the previous chapter, the reference extremity was fastened to the ground and, therefore, the *heel off* phase was approximated. Using the augmented model makes it possible to observe the exact moment where each phase occurs. Furthermore, the projection of the pelvis begins to increase again.

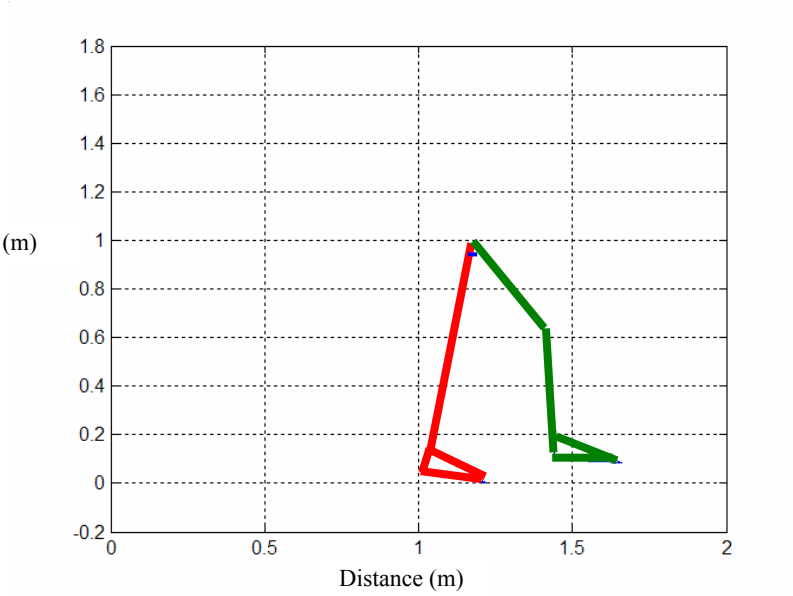


Figure 32. Heel Off

In Figure 33, a double support phase occurs for the second time. This is the moment when the reference extremity changes from one leg to another. The toe of the previous reference extremity is ready to leave the ground, and the heel of the new reference extremity strikes the ground. This phase is called *toe off*. The length of the projection of the pelvis reaches its maximum value once more.

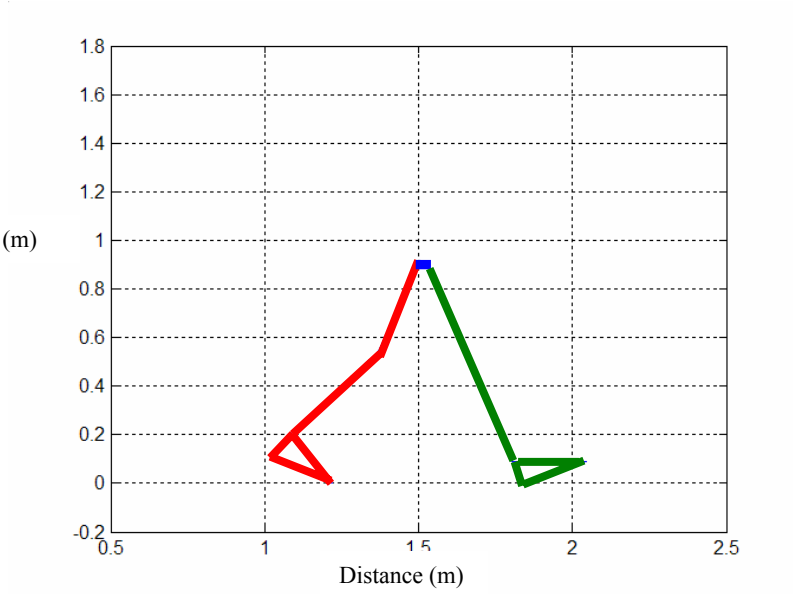


Figure 33. Toe Off

In Figure 34, the augmented model is in the *swing* phase. The foot of the supporting extremity is in full contact to the ground, while the swinging extremity is exactly under the body.

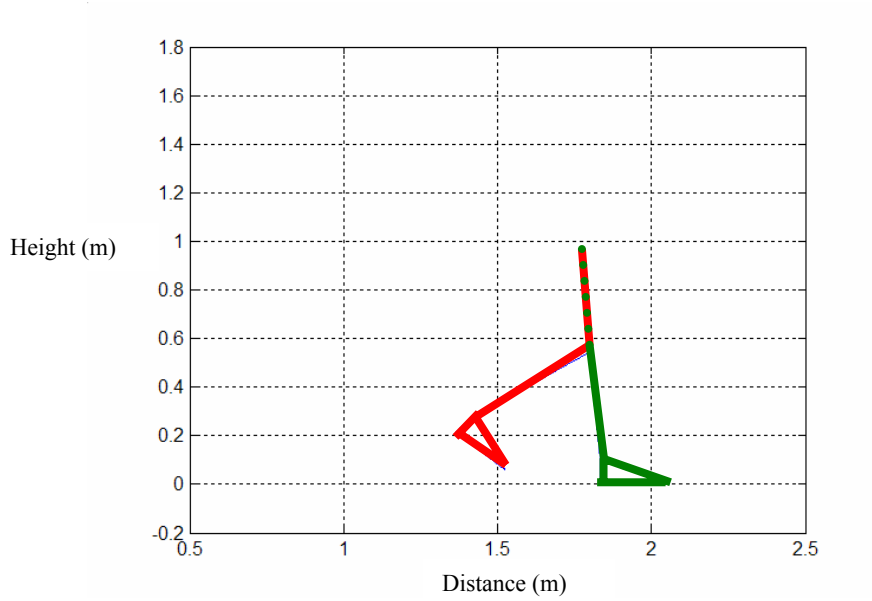


Figure 34. Acceleration

In Figure 35, the *midswing* occurs. In *midswing,* the swinging extremity passes directly beneath the body.

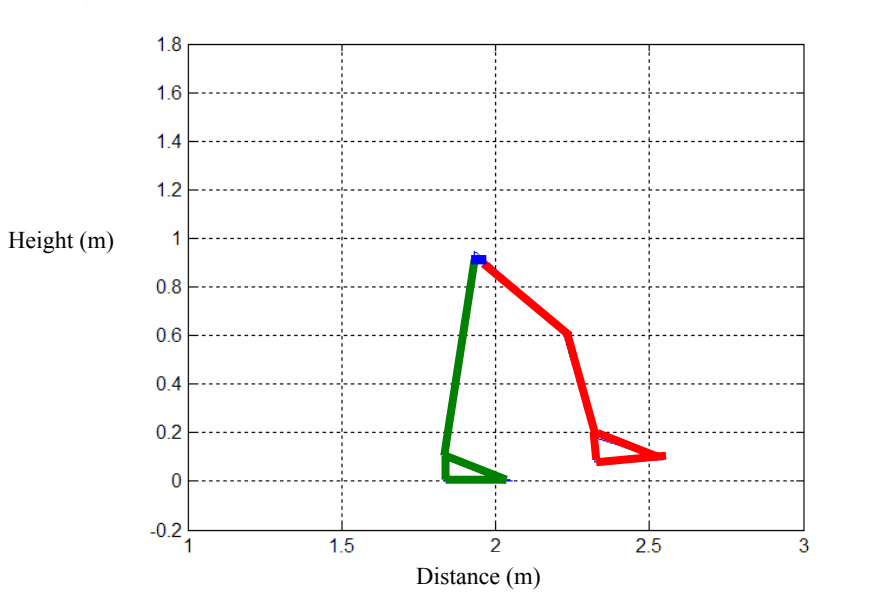


Figure 35. Midswing

In Figure 36, the last step of the gait–cycle occurs. It is called *deceleration* and it describes the moment where the knee of the swinging extremity is extending, and the heel is ready to strike the ground and initiate a new gait–cycle.

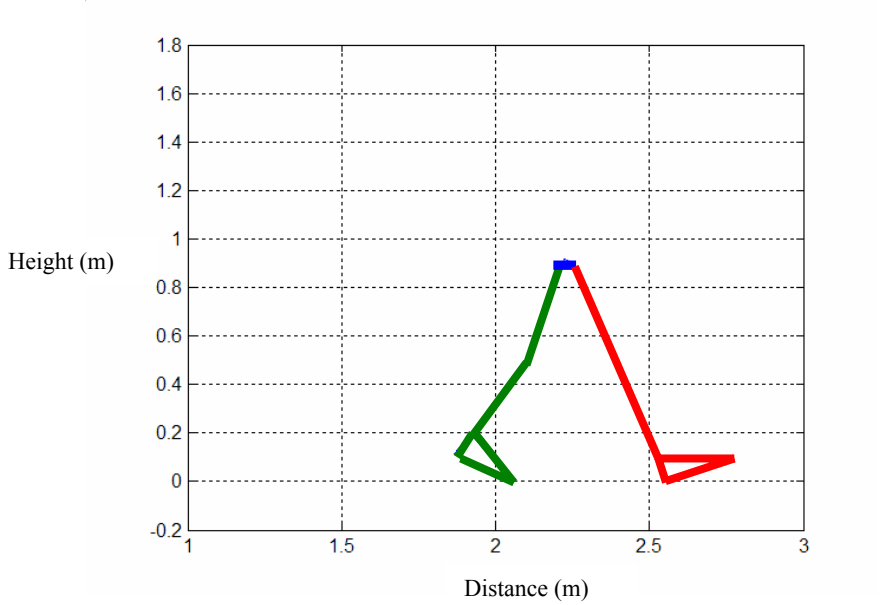


Figure 36. Deceleration

This section presented a more sophisticated simulation of the gait–cycle. The model consisted of seven segments. Two thighs, two shanks, two feet, and a pelvis. Next, a simulation of the human gait–cycle is presented in LISP.

C. A DIFFERENT APPROACH USING LISP

Another program written in LISP describes the movement of the foot during the human gait–cycle. LISP was chosen from among the other computer languages for this approach because it still is a versatile language and can be successfully applied to computer simulation of physical systems such as the human body [Ref. 32]. Professor R. McGhee provided the LISP program, which is illustrated in Appendix D.

The following figures illustrate the position of the foot during the gait–cycle. More specifically, Figure 37 depicts the foot in *heel strike*, Figure 38 refers to *foot flat*, and Figure 39 illustrates the *toe off*.

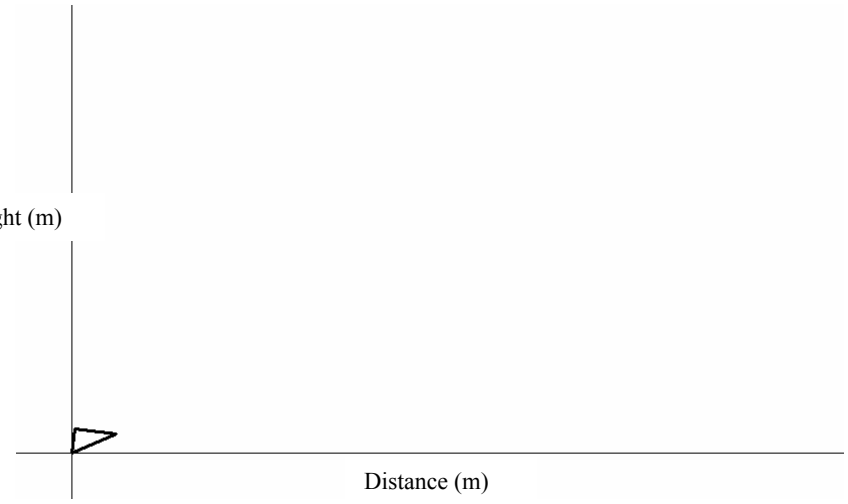


Figure 37. Heel Strike

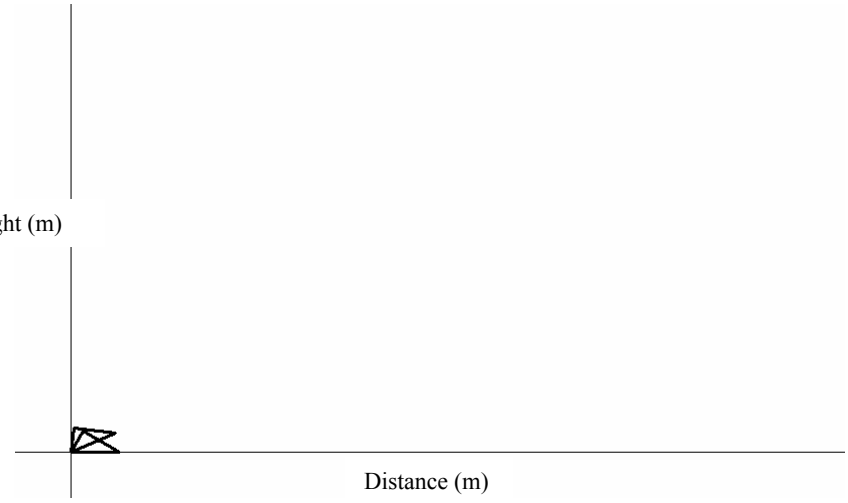


Figure 38. Foot Flat

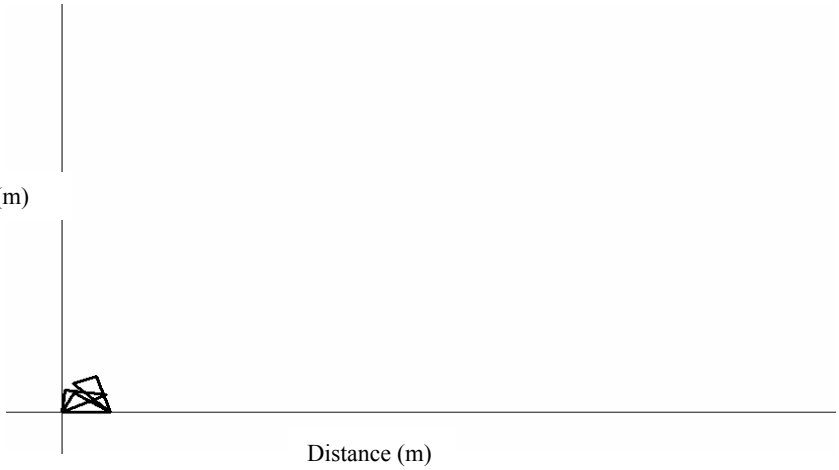


Figure 39. Toe Off

Toe off is the point where the foot leaves the ground and it passes from the *stance* phase to the *swing* phase. Therefore, Figures 40, 41, and 42 illustrate the *acceleration, midswing,* and *deceleration*, respectively.

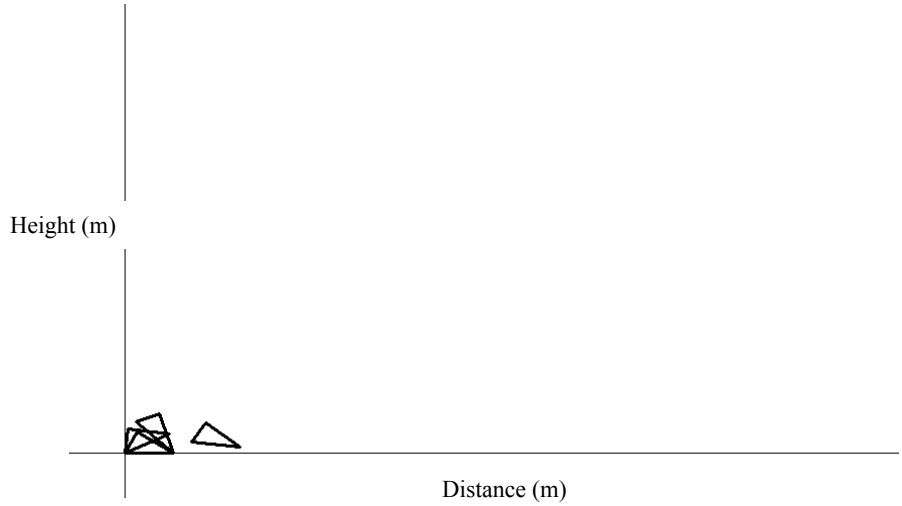


Figure 40. Acceleration

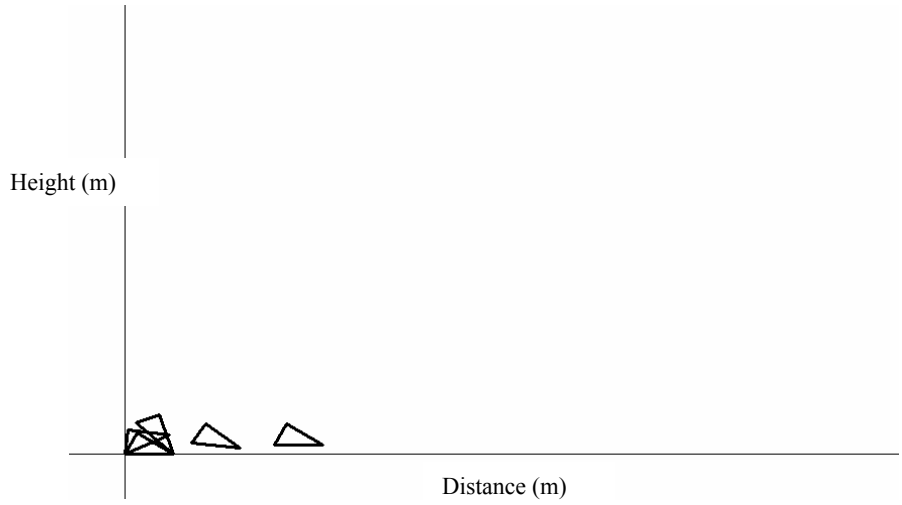


Figure 41. Midswing

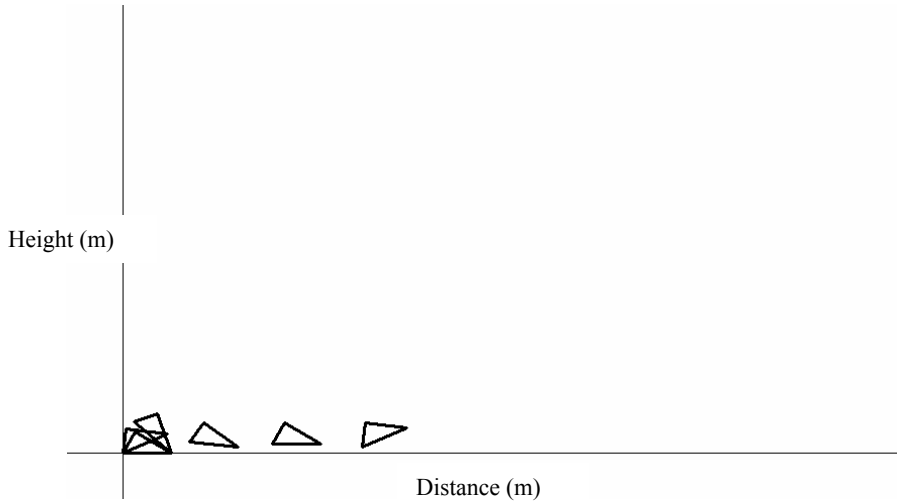


Figure 42. Deceleration

Immediately after the *deceleration* phase depicted in Figure 42, the gait–cycle starts from the beginning. This program was written in such a way so that two or more consecutives gait–cycles could be provided. Figure 43 illustrates a two–gait–cycle movement, and Figure 44 depicts a three–gait–cycle movement.

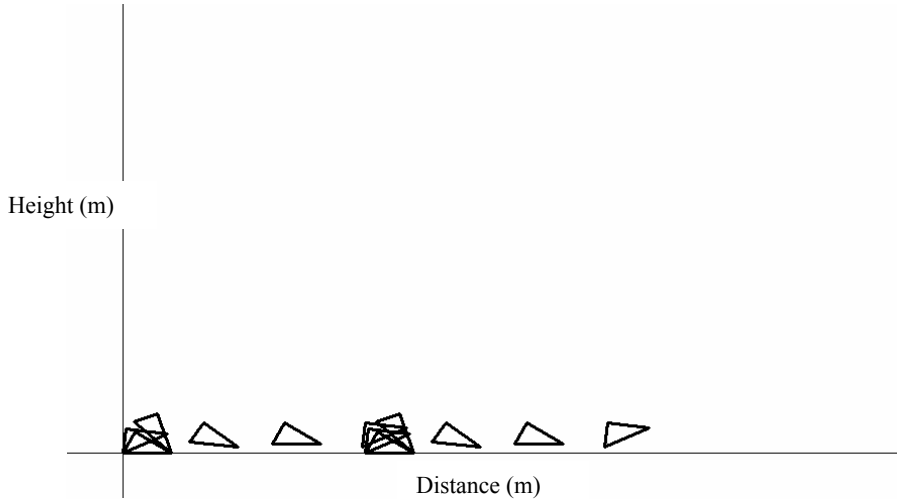


Figure 43. Two–Gait–Cycle Movement

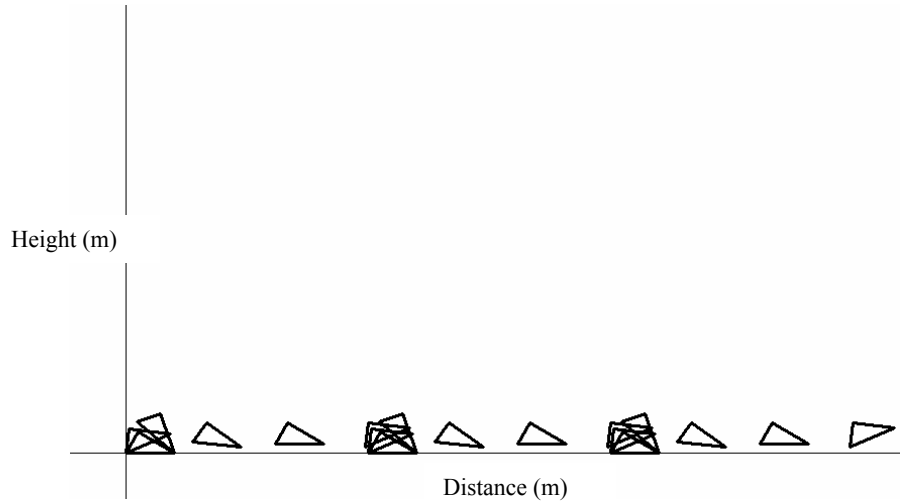


Figure 44. Three–Gait–Cycle Movement

Next, a brief comparison between the two methods is in order. Both methods describe the human motion and, particularly, the movement of the lower extremities. However, with the first approach, the movement of the entire lower human body is under study: the behavior of the pelvis in the sagittal plane, the relationships between the shanks and the thighs, and the way that the mass rigid feet are connected to the ankles. With the second approach (LISP), only the movement of one foot is under study. Furthermore, the Matlab program allows the model to move contiguously towards a certain direction. After compiling the program a window appears, and the model executes the gait–cycle walking towards the *X*–axis. This does not happen with LISP where the presentation of the gait–cycle is discrete. Thus, for each phase of the foot to be presented, a different command is given.

On the other hand, Matlab programs are considered rather difficult for someone to understand and even more difficult to intervene, compared to LISP programs, which are relatively easy to read since it has a free coding form and a simple syntax. Hence, it is more like reading a foreign language than a computer language [Ref. 32].

D. SUMMARY

This chapter discussed the implementation of an enhanced model to simulate the human gait–cycle. The differences in this model compared to that described in the previ-

ous chapter were the addition of the pelvis, and the addition of one mass rigid foot, described as a triangle, to each leg. Several figures from various phases of the simulation were illustrated.

Next, another simulation was provided by a program written in LISP. The model consisted of only one mass rigid foot, described as a triangle, and the phases of the gait–cycle as well as the outcomes of more than one gait–cycle were presented in several figures. Finally, a brief comparison between the two approaches followed.

The next chapter presents the concepts of motion tracking and position estimation. The MARG sensors were used to derive real data and to incorporate them into the Matlab program developed earlier in this project to describe a human walking.

VI. TESTING OF THE SIMULATION MODEL USING MARG SENSOR DATA

The previous chapter presented an advanced simulation of the human gait–cycle. An augmented model was implemented for that purpose. In addition to the simple model demonstrated in Chapter IV, the advanced model consisted of two feet and a pelvis. Thus, a more realistic representation of human walking was illustrated. Furthermore, the simulation for the advanced model was compared to a second simulation, in which a LISP program was used to represent the phases of a single foot during the gait–cycle.

This chapter discusses the concepts of motion tracking and position estimation. Next, the implementation of the MARG sensors to efficiently describe the motion of the lower extremities is discussed. Real data is derived and used in the existing Matlab program to describe the human gait–cycle and to estimate the position of the model.

A. MOTION TRACKING AND POSITION ESTIMATION

Tracking human motion and, especially, human walking has been the field of study for many researchers. Several techniques have been developed for that purpose and quite a few papers have been written. The following three steps describe the most conventional approach of tracking human motion [Ref. 33]:

- Derivation of real data from real human motion by using three–dimensional motion–capturing systems such as cameras, etc.
- Simulation of human motion, using theoretical approaches such as forward or inverse kinematics/dynamics.
- Interpolation between frames.

As is known, the complex nature of the human body does not allow accurate human motion tracking based on relatively simple models. There is a tradeoff between the accuracy of the result and the simplicity of the model used for that purpose. Most researchers prefer to choose simple models rather than deal with complex algorithms that increase the computational load exponentially, although current computer capabilities allow the manipulation of an extreme amount of data. The most common model is that of each limb represented by a rigid bar connected to its neighbors with a joint. Jia-Ching Cheng and Jose M. F. Moura [Ref. 17] track human walking using such a model. The model was given real data derived from a video sequence. A similar procedure was pre-

sented by A. G. Bharatkumar, K. E. Daigle, M. G. Pandy, Qin Cai and J. K. Aggarwal [Ref. 15]. Real data was obtained from people walking with markers on their limbs. Next, the real data is provided to a model similar to that described above.

Position estimation is another issue and it is not necessarily connected to tracking human motion in detail. Steven Feiner, Blair McIntyre, Tobias Hollerer, and Anthony Webster [Ref. 34] achieved a satisfactory estimation of position by using a digital GPS, magnetometers, and an inertial head tracker. However, this is good only for outside environments. For indoors environments, this configuration is rather inadequate due to the low signal level of the GPS. Masakatsu Kurogi and Takeshi Kurata [Ref. 35] proposed a method of personal positioning, which allows both indoor and outdoor motion. This method is based on the estimates of relative displacement, which used the analysis of human walking behavior using self–contained sensors. The method is also based on the estimates of position and orientation within a Kalman filtering framework.

However, a different approach in tracking human motion and, especially, human walking and estimating the position is proposed in this thesis. The positions of the limbs of the lower extremities can be estimated using the MARG sensors. They are small inertial/magnetic sensors, consisting of three accelerometers, three magnetometers, and three angular rate sensors. All three triads are orthogonally mounted [Ref. 36]. The sensors' data is processed by a filtering algorithm capable of tracking in all orientations. The set of angles provided by the algorithm is used in the Matlab program, presented in previous chapters, in order to describe human walking. Therefore, the real data provided by the sensors is implemented in the program instead of the pseudo–data. Moreover, the total distance traveled can also be calculated. Since the motion occurs in the sagittal plane, the term position estimation refers to the total distance traveled along one axis.

B. EQUIPMENT SETUP

The data acquisition was achieved by using the magnetic, angular rate and gravity (MARG) sensors. Data from the sensors is sent to the control interface unit (CIU) and then transmitted through a wireless system. Andreas Kavousanos-Kavousanakis [Ref. 2] thoroughly described, tested, and evaluated this configuration. Therefore, only a brief summary of each part is provided in this thesis. Even though MARG IV sensors were available, due to some technical difficulties concerning communication problems be-

tween the sensors themselves and the new 16–channel CIU, the previous model of MARG III sensors had to be used. The sensor consists of three magnetometers, three accelerometers, three angular rate sensors, and a microcontroller. Figure 45 illustrates a MARG III sensor.

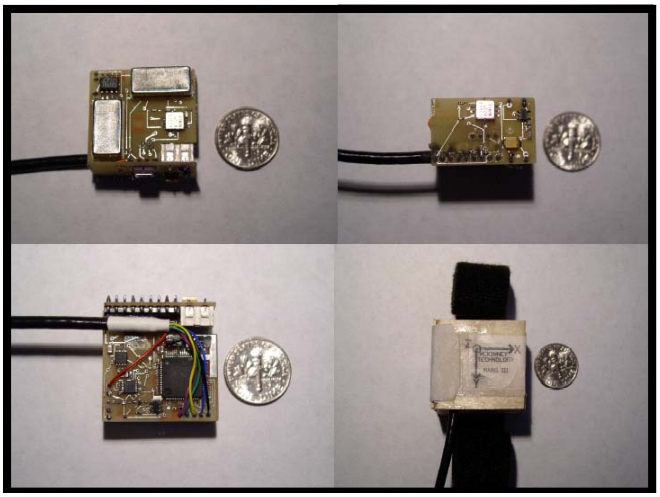


Figure 45. The MARG III Sensor [From Ref. 2.]

The data obtained from the sensors must be transmitted wirelessly to the central terminal to be further processed. Every sensor has nine elements with data flowing from each one of them. Therefore, for this kind of data to be sent across through a single channel, it must be multiplexed. This is the task of the CIU. The first CIU was a one–channel CIU. However, for the purposes of this thesis, a three–channel CIU was used, which is nothing more than three one–channel CIUs in a parallel configuration. Figure 46 illustrates a three–channel CIU.

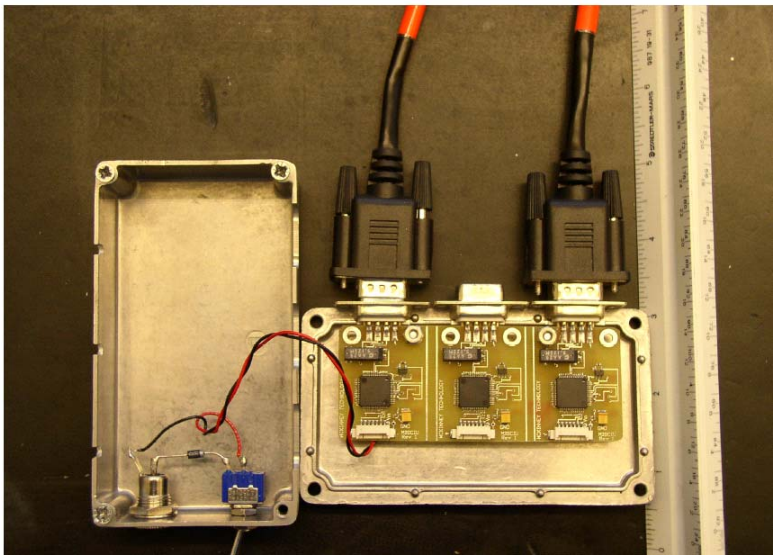


Figure 46. The Three–Channel CIU [From Ref. 2.]

During the time that this thesis was written, the 16–channel CIU was available. This CIU is capable of multiplexing the data from 16 different sensors and deliver it through a single communication channel to the terminal. However, due to the nature of the research, the use of this particular CIU was not necessary. The use of the three–channel CIU proved to be more than adequate.

C. PROCEDURE/RESULTS

As mentioned previously, the sensors generate data which were delivered to the CIU. In the CIU, the data is multiplexed in order to be transmitted wirelessly through a single channel to the terminal. The data which reach the terminal are neither understandable nor easy to use directly in other applications. Table 6 illustrates some portion of the "raw" data delivered from the CIU.

0	1	1570	1549	1500	1349	2169	1875	1903	1619	1344
0	2	1569	1548	1499	1349	2166	1876	1901	1620	1345
0	3	1569	1550	1500	1351	2170	1873	1903	1620	1345
0	4	1569	1549	1500	1350	2170	1874	1902	1619	1346
0	5	1569	1548	1500	1351	2171	1872	1903	1619	1345
0	6	1569	1548	1500	1353	2171	1872	1904	1620	1345
0	7	1569	1549	1500	1352	2174	1872	1902	1619	1344
0	8	1569	1549	1500	1353	2172	1872	1902	1620	1344
0	9	1568	1548	1500	1352	2171	1872	1902	1619	1345
0	10	1569	1548	1500	1353	2170	1874	1901	1619	1345
0	11	1570	1548	1500	1351	2170	1872	1902	1620	1346
0	12	1570	1550	1500	1353	2171	1871	1903	1619	1345
0	13	1570	1549	1500	1354	2169	1872	1902	1618	1345
0	14	1570	1548	1500	1351	2170	1871	1902	1617	1347
0	15	1570	1549	1500	1353	2168	1873	1902	1619	1345
0	16	1570	1549	1500	1352	2168	1875	1902	1618	1345
0	17	1569	1548	1500	1352	2172	1875	1903	1618	1345
0	18	1569	1548	1501	1348	2168	1875	1902	1618	1344
0	19	1568	1549	1500	1352	2169	1873	1901	1618	1345
0	20	1569	1549	1500	1351	2170	1872	1903	1619	1344

Table 6. Data Delivered from the CIU.

The first column represents the time in seconds when the specific measurement occurs (e.g., at zero seconds in Table 6). The second column is the number of the sample. There are 100 samples taken per second. The following nine columns represent the triads of the magnetometers, the accelerometers, and the angular rate sensors, respectively.

The "raw" data must be transformed. For that purpose, Andreas Kavousanos-Kavousanakis [Ref. 2] proposed the *Quest* algorithm, which used quaternions to calculate the orientation of the limb on which the sensor was attached. However, this method did not use the measurements of the angular rate sensors, which resulted in an inadequate representation of the human motion during large accelerations. To fix this problem, Conrado Aparicio [Ref. 3] presented the *Factored Quaternion* algorithm. A Kalman filter was implemented and combined the static estimated quaternion with the measurements from the angular rate sensors to provide the optimal orientation estimate. Both Andreas Kavousanos-Kavousanakis and Conrado Aparicio managed to track the motion of the right arm in real time.

However, the main objective of this thesis was not to track the motion of the lower extremities in real time. The goal was to gather data from the sensors and to incorporate it into the existing Matlab's simulation program to move the model. As mentioned previously, the three–channel CIU is used for processing the data obtained from the sensor. Another issue is that the existing software does not allow taking data from more than one sensor at the same time. Thus only one sensor is used. For example, the sensor was attached to the thigh and after running the program, data were obtained and saved in a text file to be loaded in the Matlab program. Next, the sensor was attached to the shank and the same procedure was followed. This, of course, creates an additional difficulty, since the samples from the two measurements to be incorporated into the Matlab program must be as highly correlated as possible.

The "raw" data delivered from the CIU is transformed into quaternions and then into angles by using the *Factored Quaternion* algorithm. Table 7 illustrates the quaternion measurements of a sensor attached to the right thigh, while the person is in the stand–up position. In the table, the first 10 samples of the experiment are shown.

Sample	$\widehat{q}_0$	$\widehat{q}_1$	$\widehat{q}_2$	$\widehat{q}_3$
1	0.29198190246990	0.66565006440052	0.30672889578596	0.61447045891854
2	0.29292186055944	0.66443412124903	0.30896746098712	0.61421754303960
3	0.29410101634827	0.66505770870310	0.30936581566533	0.61277698094431
4.	0.29514747448117	0.66518018298469	0.30958597513655	0.61202926112286
5	0.29590692455467	0.66564386890250	0.30970296314723	0.61109852430761
6	0.29672858201358	0.66595294873036	0.31003756145395	0.61019302616568
7	0.29747698061030	0.66644882946701	0.30977408729708	0.60942055966984
8	0.29820349632252	0.66664910321432	0.30994922099631	0.60875703558016
9	0.29887764092890	0.66675166731840	0.31025910833958	0.60815594675481
10	0.29944922658029	0.66677150254690	0.31053376802901	0.60771268129398

Table 7. Quaternion Measurements for Sensor Attached to Right Thigh in the Stand–Up Position.

The *Factored Quaternion* algorithm also transforms the quaternions into angles. Table 8 illustrates the angle measurements of a sensor that is attached to the right thigh, while the person is in the stand–up position. More specifically, the first 10 samples of the experiment are shown.

Sample	*X*–axis (Roll)	*Y*–axis (Pitch)	*Z*–axis (Yaw)
1	171.008478205228	85.683252083344	41.139346835021
2	172.512741690264	85.727367806572	42.927526774052
3	170.997251482504	85.585802676671	41.555987864367
4	169.861075130236	85.536055177006	40.534336684944
5	168.817169040274	85.439891718533	39.570874573482
6	167.994819515943	85.353669467500	38.853972564782
7	166.716323701608	85.271536035443	37.620624398289
8	165.975826477322	85.212126041704	36.962298928267
9	165.454142783164	85.161905494336	36.530052334811
10	165.053655385336	85.129530808321	36.206296216028

Table 8. Angle Measurements for Sensor Attached to Right Thigh in the Stand–Up Position.

At this point, the following question arises. Since the person is standing up and facing the magnetic north during the measurement, what is the reason for the deviation of the measured values from the theoretical ones? For example, the *roll* value is approximately 171 degrees instead of 180 and the *pitch* is 85.7 degrees instead of the theoretical 90. One reason is definitely human error in aligning the sensor to the magnetic north. Another reason is the slight miscalculations that might have occurred during the initial calibration of the sensor. However, after conducting several tests in which the axes of the sensor were completely aligned to the Earth's fixed coordinate system, the following remarks are provided.

The measurements for the *roll* vary from –4 to 4 degrees with zero being the theoretical value. Similarly, the measurements for the *pitch* vary from 4 to 9 degrees with zero being the theoretical value. Finally, the measurements for the *yaw* vary from –4 to –8 degrees with zero being the theoretical value. Those deviations are taken into consideration during the phase where exhaustive tests are performed to collect data for the Matlab program. However, the most important reason is that in an indoor environment, the various electromagnetic fields affect the magnetometers of the sensors resulting in this rather significant deviation.

During the experiments, only one sensor was utilized. The sensor was already calibrated and attached in such a way so that the *Y*–axis would point to the magnetic north. Furthermore, the *X*–axis would point to the head of the model and the *Z*–axis would be consistent with the right hand rule. Therefore, in the simulation, the model is moving towards the *Y*–axis. The sensor was first attached to the supporting thigh, then to the supporting shank, then to the swinging thigh, and finally to the swinging shank. Measurements were taken separately from each limb and many sets of real data were saved to be implemented to the simulation program. The initial position of the model, as illustrated in Figure 47, depicts the supporting limbs in the *toe off* position and the swinging limbs in the *foot flat* position.

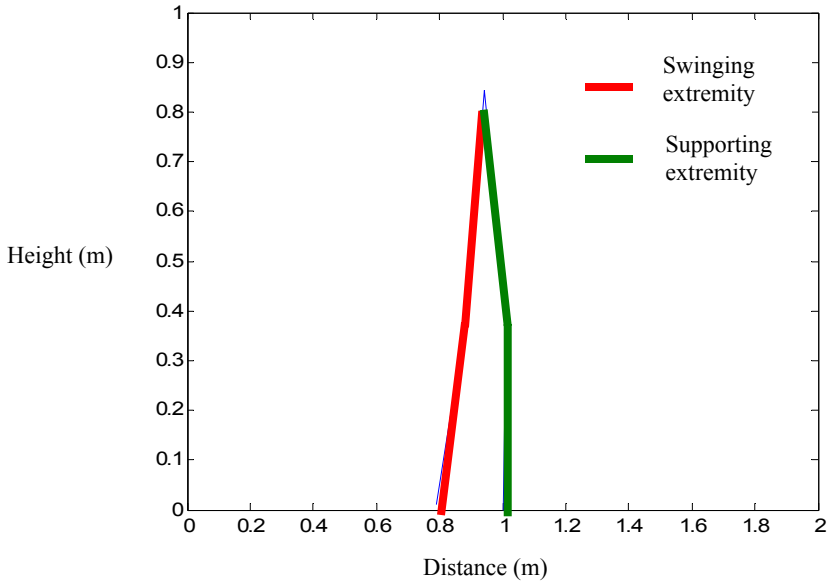


Figure 47. Initial Position of the Model during the Real Data Simulation

Since only one sensor could be used at each time, the movement of the model had to be exactly the same for the extracted sets of data to be consistent. However, it is more than obvious from the simulation that, no matter how well correlated are the data, it will never have as satisfactory results as when four individual sensors are attached to the four limbs and the data is processed simultaneously.

The Matlab program, provided in Appendix E, consists of two parts. The first part is the *Factored Quaternion* algorithm, which derives the values of the angles in all directions. The second part implements the angles in a simulation to track the motion of the lower extremities. Figure 48 illustrates the final position of the model. The model is at the *double support* phase.

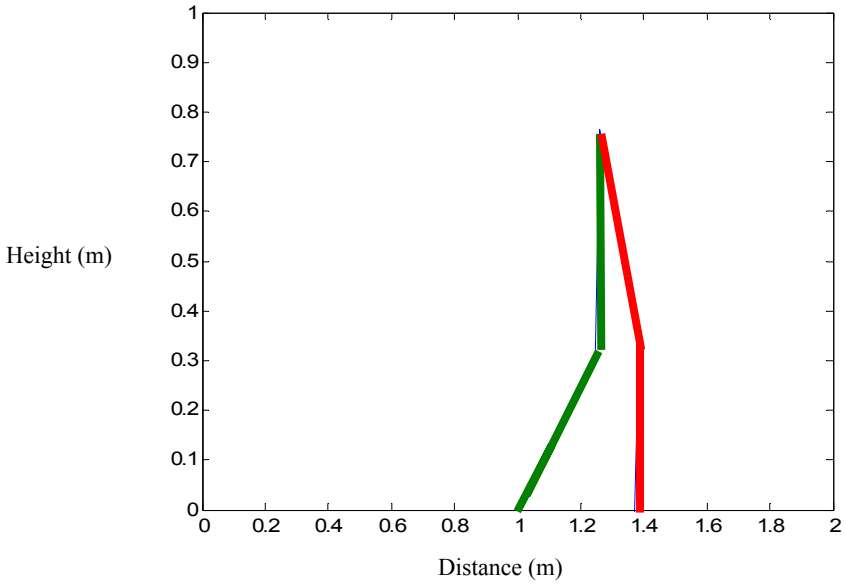


Figure 48. Final Position of the Model in Real Data Simulation

During the experiments, the actual length of the step was measured and found to be 84 cm. The program is not only able to track the motion of the lower extremities, but to estimate the position of the model as well. Since the motion of the model occurs in the sagittal plane, the estimation of the position of the model automatically translates to the estimation of the distance that the model has traveled in the *Y*–axis. From the experiments conducted, the mean value of the step length was found equal to 60 cm. However, if the length of the foot is also added, then the length of the step reaches 82 cm, which agrees with the estimated value.

The following remark deals with the position of the model when it moves in three dimensions and it might be useful information for moving this project to another level. As mentioned previously, the *Y*–axis of the attached sensor points to the magnetic north. The *yaw* angle is equal to 90 degrees. When the sensor is turned towards East, the *yaw* angle goes all the way up to 180 degrees. On the other hand, when the sensor is turned towards West, the *yaw* angle goes all the way up to zero degrees. Therefore, if the simulation program is modified so that the model undergoes a three–dimension movement, the transition from the sagittal to the frontal plane and vice versa, could be identified by the values of the *yaw* angles.

D. SUMMARY

This chapter discussed the concepts of tracking human motion and position estimation. Next, the implementation of the MARG III sensors in tracking the motion of the lower extremities and estimating the position of the model is discussed. Furthermore, the equipment utilized in this project was presented, along with a brief description. Finally, the procedure for processing the data from the sensors for the simulation program was discussed and the results of the simulation, where real data were used, were illustrated.

The next chapter presents the conclusions and the contributions of this thesis. It also provides suggestions for further improvements and future work.

VII. CONCLUSIONS AND FUTURE WORK

A. CONCLUSIONS

This thesis was able to track the motion of the lower extremities of a human and to estimate the traveled distance. The MARG sensors were used to track the motion of the lower extremities and estimate the position of the model. The lower extremities were modeled as an articulated object, which consisted of rigid bars connected to each other by joints.

The goal of this thesis was accomplished by incorporating real data into a computer simulation written in Matlab. The simulation model was initially driven by pseudo–data. The pseudo–data was nothing more than a set of angles which, after being given an initial value, were incremented or decremented, depending on the limb and on the phase of the gait–cycle. To accurately simulate human walking, the basic principles of biomechanics as well as the phases of the gait–cycle had to be taken into consideration in the formation of the pseudo–data. Furthermore, a mathematical model, based on the theory of forward kinematics and the theory of manipulators kinematics, was derived. This model formed the main part of the Matlab program, and the pseudo–data, as well as the real data, were incorporated directly in it.

The real data was derived from a magnetic, angular rate, and gravity (MARG) sensor. For the purposes of this thesis, a MARG III sensor was used. The data derived from the sensor was processed by a control interface unit (CIU). The CIU multiplexed the data so that it could go through a single channel and be transmitted wirelessly to the central terminal. Next, it was further processed by the *Factored Quaternion* algorithm, which transformed the "raw" data into quaternions and then into angles, calculating the orientation in every direction. The set of angles was provided to the simulation model, which was then able to track the motion of the lower limbs efficiently.

The existing software did not allow the incorporation of data from more than one sensor at the same time. Therefore, for example, in order to track the motion of one leg consisting of a thigh and a shank, the sensor first had to be attached to the thigh, and provide the data related to the motion of the thigh. Next, it had to be attached to the shank

and provide the data related to the motion of the shank. Obviously, the outcome was not as good as it would have been if the measurements had been taken from both locations at the same time. In order to overcome that difficulty and to provide a result that would be as realistic as possible, extensive tests had to be conducted. The measurements that provided the best correlation between the thigh data and the shank data were incorporated into the Matlab simulation program. The estimation of the position of the model in a two–dimensional environment translates to the estimation of the distance that the model has traveled along one axis. From the experiments conducted, the mean value of the step length was found equal to 82 cm, which agrees with the estimated value (84 cm).

Another important remark is that this thesis did not deal with real time human motion tracking. The objective of this thesis was to study the implementation of the MARG III sensors in tracking the motion of the lower extremities by incorporating the pseudo–data or the data obtained from the sensors into a simulation program written in Matlab. There was no implementation of any algorithm in Java whatsoever, unlike some of the references.

In addition, the various measurements of the sensors showed that a significant deviation existed between the measured values of the angles and the theoretical values, possibly due to errors during the original calibration. However, the existence of various electromagnetic fields, which definitely affected the magnetometers of the sensors, seems to be a more likely explanation for this deviation.

B. FUTURE WORK

This thesis achieved the tracking of the motion of the lower extremities. However, to reach a point to which all of this can be implemented in a virtual environment, much more remains to be done.

The possibility of obtaining data from more than one sensor is absolutely essential for the completion of the MARG project. As far as the lower extremities are concerned, the utilization of six sensors would be enough. Two of them would be attached on the thighs, two on the shanks, and two on the feet. The derivation of data from all the sensors at the same time would prevent the researchers from conducting exhaustive experiments to achieve the best correlation between the data.

Another issue that must be addressed is the communication problems between the MARG IV sensors and the 16–channel CIU that appeared during the research. The utilization of the 16–channel CIU will allow the depiction of the motion of all the limbs to which a sensor is attached.

The simulation model used in this thesis restricted movement to the sagittal plane. Consequently, the position estimation of the model refers to the estimation of the distance that the model traveled in one direction. The next step is to move this concept to a higher level by developing a model that would move in three dimensions instead of two. Hence, the term position estimation would refer to the position of the model in an extended horizontal plane as defined in Chapter II.

Finally, tracking the motion of the lower extremities in real time would be very important as well.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A

This Appendix contains the Matlab program which tested the mathematical model of the human–gait. (Comments are highlighted in green.)

```
% LTJG (M) I. PANTAZIS (HN)
% 09-01-2005
% FORWARD KINEMATICS TEST
% In this test we are going to verify that the background theory that we
% introduced and the formulas that we have developed apply for random joint angles.

clear
clc
alpha_0=0;, a0=0;, d1=0;, theta_1=pi/3;
l1=0.45;                                        % Note that the lengths are in meters
alpha_1=0;, a1=l1;, d2=0;, theta_2=pi/6;
T_01=[cos(theta_1),-sin(theta_1),0,0;sin(theta_1),cos(theta_1),0,0;0,0,1,0;0,0,0,1];
T_12=[cos(theta_2),-sin(theta_2),0,l1;sin(theta_2),cos(theta_2),0,0;0,0,1,0;0,0,0,1];
T_02=T_01*T_12;
l2=0.4;
P2=[l2;0;0;1];
P_02=T_02*P2

% Since theta_1 plus theta_2 is equal to pi/2 and since we consider this
% model to be two-dimensional, the projection of the pelvis axis to the
% x-axis is going to be almost parallel to the ground. Therefore
% theta_3=3*pi/2.

theta_3=3*pi/2;
T_23=[cos(theta_3),-sin(theta_3),0,l2;sin(theta_3),cos(theta_3),0,0;0,0,1,0;0,0,0,1];
T_03=T_02*T_23;
l_pelvis=0.2;
phi=pi/60;
P3=[l_pelvis*sin(phi);l_pelvis*sin(-phi);0;1];
P_03=T_03*P3
theta_4=5*pi/3;
alpha_3=0;
a3=l_pelvis*sin(phi);
d4=0;
T_34=[cos(theta_4),-sin(theta_4),0,l_pelvis*sin(phi);sin(theta_4),cos(theta_4),0,0;0,0,1,0;0,0,0,1];
T_04=T_03*T_34;
P4=[l2;0;0;1];
P_04=T_04*P4
theta_5=-1.5*pi/18;
alpha_4=0;
a4=l2;
d5=0;
T_45=[cos(theta_5),-sin(theta_5),0,l2;sin(theta_5),cos(theta_5),0,0;0,0,1,0;0,0,0,1];
T_05=T_04*T_45;
P5=[l1;0;0;1];
P_05=T_05*P5
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B

This Appendix contains the Matlab simulation program for the case presented in Chapter IV. (Comments are highlighted in green.)

```
% 21 FEB 2005
% LTJG (M) PANTAZIS IOANNIS (HN)
% ANIMATION - FOUR LIMPS (NO PELVIS)- FULL GAIT CYCLE-CHANGEABLE STEP-TREADMILL

clear
clc
l1=0.45;
l2=0.4;
l3=l2;
l4=l1;
x_0=1;
y_0=0;
theta_1_initial=-(pi/6);
theta_2_initial=-(pi/6);
theta_3_initial=-(11*pi/12);
x_5=x_0+l1*sin(theta_1_initial)+l2*sin(theta_2_initial)+l3*sin(theta_3_initial)-sqrt((l1^2)-(((l1+l2)*cos(theta_1_initial)+l3*cos(theta_3_initial))^2));
x=[x_0,x_0+l1*sin(theta_1_initial),x_0+l1*sin(theta_1_initial)+l2*sin(theta_2_initial),x_0+l1*sin(theta_1_initial)+l2*sin(theta_2_initial)+l3*sin(theta_3_initial),x_5];
y=[y_0,l1*cos(theta_1_initial),(l1+l2)*cos(theta_1_initial),(l1+l2)*cos(theta_1_initial)+l3*cos(theta_3_initial),0];
nframes=50;
h=plot(x,y,'m')
set(h,'MarkerSize',18);
axis([-1 8 0 4])
axis square
grid
k1=10;
k2=20;
k3=26;
k4=30;
k5=37;
k6=45;
for i=1:4
for k = 1:nframes
if k<=k1
k
alpha=(100-rand)/100
theta_1_inc=alpha*pi/60;
theta_2_inc=pi/60;
theta_3_a_inc=pi/180;
theta_4_a_inc=pi/150;
theta1=theta_1_initial+k*theta_1_inc;
theta2=theta_2_initial
theta3=(theta_3_initial-k*theta_3_a_inc)
```

```
        theta4_constant=-(pi/2)-atan((0-((l1+l2)*cos(theta_1_initial)+l3*cos(theta_3_initial)))/(x_5-
(x_0+l1*sin(theta_1_initial)+l2*sin(theta_2_initial)+l3*sin(theta_3_initial))));
        theta4=theta4_constant+k*theta_4_a_inc;
        x_init=[x_0,x_0,x_0,x_0,x_0];
        x(i+1,:)=x(i,1)+[0,l1*sin(theta1),l1*sin(theta1)+l2*sin(theta2),l1*sin(theta1)+l2*sin(theta2)+l3*si
n(theta3),l1*sin(theta1)+l2*sin(theta2)+l3*sin(theta3)+l4*sin(theta4)]
        X=x(i+1,:)
        y=[y_0,l1*cos(theta1),l1*cos(theta1)+l2*cos(theta2),l1*cos(theta1)+l2*cos(theta2)+l3*cos(theta3
),l1*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l4*cos(theta4)]
        elseif  k<=k2
        k
        theta_3_b_inc=pi/60;
        theta_4_b_inc=pi/60;
        theta1=theta_1_initial+k1*theta_1_inc;
        theta2=theta_2_initial+(k-k1)*theta_2_inc;
        theta3=theta_3_initial-k1*theta_3_a_inc-(k-k1)*theta_3_b_inc;
        theta4=theta4_constant+k1*theta_4_a_inc-(k-k1)*theta_4_b_inc;
        x(i+1,:)=x(i,1)+[0,l1*sin(theta1),l1*sin(theta1)+l2*sin(theta2),l1*sin(theta1)+l2*sin(theta2)+l3*si
n(theta3),l1*sin(theta1)+l2*sin(theta2)+l3*sin(theta3)+l4*sin(theta4)]
        X=x(i+1,:)
        y=[y_0,l1*cos(theta1),l1*cos(theta1)+l2*cos(theta2),l1*cos(theta1)+l2*cos(theta2)+l3*cos(theta3
),l1*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l4*cos(theta4)]
        elseif     k<=k3
        k
        theta_3_c_inc=pi/80;
        theta_4_c_inc=pi/90;
        theta1=theta_1_initial+(k-k1)*theta_1_inc;
        theta2=theta_2_initial+(k-k1)*theta_2_inc;
        theta3=theta_3_initial-k1*theta_3_a_inc-(k2-k1)*theta_3_b_inc-(k-k2)*theta_3_c_inc;
        theta4=theta4_constant+k1*theta_4_a_inc-(k2-k1)*theta_4_b_inc-(k-k2)*theta_4_c_inc;
        x(i+1,:)=x(i,1)+[0,l1*sin(theta1),l1*sin(theta1)+l2*sin(theta2),l1*sin(theta1)+l2*sin(theta2)+l3*si
n(theta3),l1*sin(theta1)+l2*sin(theta2)+l3*sin(theta3)+l4*sin(theta4)]
        X=x(i+1,:)
        y=[y_0,l1*cos(theta1),l1*cos(theta1)+l2*cos(theta2),l1*cos(theta1)+l2*cos(theta2)+l3*cos(theta3
),l1*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l4*cos(theta4)]
        elseif k<=k4
        k
        theta_2_d_inc=pi/90;
        theta_3_d_inc=pi/145;
        theta_4_d_inc=pi/47;
        theta1=theta_1_initial+(k3-k1)*theta_1_inc+(k-k3)*theta_1_inc;
        theta2=theta_1_initial+(k3-k1)*theta_1_inc-(k-k3)*theta_2_d_inc;
        theta3=theta_3_initial-k1*theta_3_a_inc-(k2-k1)*theta_3_b_inc-(k3-k2)*theta_3_c_inc+(k-
k3)*theta_3_d_inc;
        theta4=theta4_constant+k1*theta_4_a_inc-(k2-k1)*theta_4_b_inc-(k3-k2)*theta_4_c_inc-(k-
k3)*theta_4_d_inc;
        x(i+1,:)=x(i,1)+[0,l1*sin(theta1),l1*sin(theta1)+l2*sin(theta2),l1*sin(theta1)+l2*sin(theta2)+l3*si
n(theta3),l1*sin(theta1)+l2*sin(theta2)+l3*sin(theta3)+l4*sin(theta4)]
        X=x(i+1,:)
        y=[y_0,l1*cos(theta1),l1*cos(theta1)+l2*cos(theta2),l1*cos(theta1)+l2*cos(theta2)+l3*cos(theta3
),l1*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l4*cos(theta4)]
        elseif  k<=k5
        k
        theta_1_e_inc=pi/40;
        theta_2_e_inc=pi/90;
        theta_3_e_inc=pi/40;
```

```
        theta_4_e_inc=pi/80;
        theta1_f=theta_1_initial+(k3-k1)*theta_1_inc+(k4-k3)*theta_1_inc;
        theta2_f=theta_1_initial+(k3-k1)*theta_1_inc-(k4-k3)*theta_2_d_inc;
        theta3_f=theta_3_initial-k1*theta_3_a_inc-(k2-k1)*theta_3_b_inc-(k3-k2)*theta_3_c_inc+(k4-
k3)*theta_3_d_inc;
        theta4_f=theta4_constant+k1*theta_4_a_inc-(k2-k1)*theta_4_b_inc-(k3-k2)*theta_4_c_inc-(k4-
k3)*theta_4_d_inc;
        x_01=[0,l1*sin(theta1_f),l1*sin(theta1_f)+l2*sin(theta2_f),l1*sin(theta1_f)+l2*sin(theta2_f)+l3*s
in(theta3_f),l1*sin(theta1_f)+l2*sin(theta2_f)+l3*sin(theta3_f)+l4*sin(theta4_f)]
        y_01=[y_0,l1*cos(theta1_f),l1*cos(theta1_f)+l2*cos(theta2_f),l1*cos(theta1_f)+l2*cos(theta2_f)
+l3*cos(theta3_f),l1*cos(theta1_f)+l2*cos(theta2_f)+l3*cos(theta3_f)+l4*cos(theta4_f)]
        theta4=theta4_f+(k-k4)*theta_4_e_inc;
        theta3=theta3_f+(k-k4)*theta_3_e_inc;
        theta2=theta2_f-(k-k4)*theta_2_e_inc;
        theta1=theta1_f+(k-k4)*theta_1_e_inc;
        x(i+1,:)=x(i,1)+[x_01(1,5)-l4*sin(theta4)-l3*sin(theta3)-l2*sin(theta2)-l1*sin(theta1),x_01(1,5)-
l4*sin(theta4)-l3*sin(theta3)-l2*sin(theta2),x_01(1,5)-l4*sin(theta4)-l3*sin(theta3),x_01(1,5)-
l4*sin(theta4),x_01(1,5)]
        X=x(i+1,:)
        y=[-l4*cos(theta4)-l3*cos(theta3)-l2*cos(theta2)-l1*cos(theta1),-l4*cos(theta4)-l3*cos(theta3)-
l2*cos(theta2),-l4*cos(theta4)-l3*cos(theta3),-l4*cos(theta4),0]
        elseif k<=k6
        k
        theta_1_g_inc=pi/30;
        theta_2_g_inc=pi/40;
        theta_3_g_inc=pi/80;
        theta_4_g_inc=pi/80;
        theta4=theta4_f+(k5-k4)*theta_4_e_inc+(k-k5)*theta_4_g_inc;
        theta3=theta3_f+(k5-k4)*theta_3_e_inc+(k-k5)*theta_3_g_inc;
        theta2=theta2_f-(k5-k4)*theta_2_e_inc-(k-k5)*theta_2_g_inc;
        theta1=theta1_f+(k5-k4)*theta_1_e_inc-(k-k5)*theta_1_g_inc;
        x(i+1,:)=x(i,1)+[x_01(1,5)-l4*sin(theta4)-l3*sin(theta3)-l2*sin(theta2)-l1*sin(theta1),x_01(1,5)-
l4*sin(theta4)-l3*sin(theta3)-l2*sin(theta2),x_01(1,5)-l4*sin(theta4)-l3*sin(theta3),x_01(1,5)-
l4*sin(theta4),x_01(1,5)]
        X=x(i+1,:)
        y=[-l4*cos(theta4)-l3*cos(theta3)-l2*cos(theta2)-l1*cos(theta1),-l4*cos(theta4)-l3*cos(theta3)-
l2*cos(theta2),-l4*cos(theta4)-l3*cos(theta3),-l4*cos(theta4),0]
        else k<=nframes
        k
        theta_1_h_inc=pi/22;
        theta_2_h_inc=pi/95;
        theta_3_h_inc=pi/1000;
        theta_4_h_inc=pi/1000;
        theta4=theta4_f+(k5-k4)*theta_4_e_inc+(k6-k5)*theta_4_g_inc+(k-k6)*theta_4_h_inc;
        theta3=theta3_f+(k5-k4)*theta_3_e_inc+(k6-k5)*theta_3_g_inc+(k-k6)*theta_3_h_inc;
        theta2=theta2_f-(k5-k4)*theta_2_e_inc-(k6-k5)*theta_2_g_inc+(k-k6)*theta_2_h_inc;
        theta1=theta1_f+(k5-k4)*theta_1_e_inc-(k6-k5)*theta_1_g_inc-(k-k6)*theta_1_h_inc;
        x(i+1,:)=x(i,1)+[x_01(1,5)-l4*sin(theta4)-l3*sin(theta3)-l2*sin(theta2)-l1*sin(theta1),x_01(1,5)-
l4*sin(theta4)-l3*sin(theta3)-l2*sin(theta2),x_01(1,5)-l4*sin(theta4)-l3*sin(theta3),x_01(1,5)-
l4*sin(theta4),x_01(1,5)]
        X=x(i+1,:)
        y=[-l4*cos(theta4)-l3*cos(theta3)-l2*cos(theta2)-l1*cos(theta1),-l4*cos(theta4)-l3*cos(theta3)-
l2*cos(theta2),-l4*cos(theta4)-l3*cos(theta3),-l4*cos(theta4),0]
        end
        set(h,'XData',X,'YData',y)
        M(k,i) = getframe;
```

```
end
end
total_distance=X(1,1)-x_5
distance_per_cycle=total_distance/i
% movie(M,3)
```

APPENDIX C

This Appendix contains the Matlab simulation program with the enhancements presented in Chapter V. (Comments are highlighted in green.)

```
% 22 APR 2005
% PANTAZIS IOANNIS
% ANIMATION - RIGHT FOOT (WITH MASS), SHANK, THIGH, PELVIS & LEFT THIGH, SHANK AND
% FOOT (MASS) - FULL GAIT CYCLE

clear
clc
lo=0.1;
l1=0.22;
l2=0.45;
l3=0.4;
l4=l3;
l5=l2;
l6=l1;
loo=lo;
l_pel=0.4;
x=[1-lo*sin(pi/6),1,1+l1*cos(pi/6),1-lo*sin(pi/6),1-lo*sin(pi/6)-l2*sin(pi/8),1-lo*sin(pi/6)-l2*sin(pi/8)-l3*sin(pi/8),1-lo*sin(pi/6)-l2*sin(pi/8)-l3*sin(pi/8)-l_pel*tan(pi/45),1-lo*sin(pi/6)-l2*sin(pi/8)-l3*sin(pi/8)-l_pel*tan(pi/45)-l4*sin(pi/10),1-lo*sin(pi/6)-l2*sin(pi/8)-l3*sin(pi/8)-l_pel*tan(pi/45)-l4*sin(pi/10)-l5*sin(pi/4.25),1-lo*sin(pi/6)-l2*sin(pi/8)-l3*sin(pi/8)-l_pel*tan(pi/45)-l4*sin(pi/10)-l5*sin(pi/4.25)-loo*sin(pi/4),1-lo*sin(pi/6)-l2*sin(pi/8)-l3*sin(pi/8)-l_pel*tan(pi/45)-l4*sin(pi/10)-l5*sin(pi/4.25)-loo*sin(pi/4)+l6*cos(pi/6),1-lo*sin(pi/6)-l2*sin(pi/8)-l3*sin(pi/8)-l_pel*tan(pi/45)-l4*sin(pi/10)-l5*sin(pi/4.25)];
y=[lo*cos(pi/6),0,l1*sin(pi/6),lo*cos(pi/6),lo*cos(pi/6)+l2*cos(pi/8),lo*cos(pi/6)+l2*cos(pi/8)+l3*cos(pi/8),lo*cos(pi/6)+l2*cos(pi/8)+l3*cos(pi/8)+l_pel*tan(pi/60),lo*cos(pi/6)+l2*cos(pi/8)+l3*cos(pi/8)+l_pel*tan(pi/60)-l4*cos(pi/10),lo*cos(pi/6)+l2*cos(pi/8)+l3*cos(pi/8)+l_pel*tan(pi/60)-l4*cos(pi/10)-l5*cos(pi/4.25),lo*cos(pi/6)+l2*cos(pi/8)+l3*cos(pi/8)+l_pel*tan(pi/60)-l4*cos(pi/10)-l5*cos(pi/4.25)-loo*cos(pi/4),lo*cos(pi/6)+l2*cos(pi/8)+l3*cos(pi/8)+l_pel*tan(pi/60)-l4*cos(pi/10)-l5*cos(pi/4.25)-loo*cos(pi/4)-l6*sin(pi/6),lo*cos(pi/6)+l2*cos(pi/8)+l3*cos(pi/8)+l_pel*tan(pi/60)-l4*cos(pi/10)-l5*cos(pi/4.25)];
nframes=480;
theta_0_inc=-3*pi/1800;
theta_1_inc=-3*pi/1800;
theta_2_a_inc=-pi/900;
theta_2_b_inc=-pi/5000;
theta_2_c_inc=-pi/1000;
theta_2_d_inc=-pi/500;
theta_3_b_inc=-pi/800;
theta_3_d_inc=pi/850;
theta_4_a_inc=-pi/1500;
theta_4_b_inc=-pi/400;
theta_4_c_inc=pi/2000;
theta_5_a_inc=pi/2000;
theta_5_b_inc=-pi/1000;
theta_5_c_inc=-pi/300;
theta_5_d_inc=-pi/1000;
```

```
theta_6_a_inc=pi/1000;
theta_pel_hor_a_inc=pi/9000;
theta_pel_ver_a_inc=pi/12000;
theta_pel_hor_b_inc=pi/4500;
theta_pel_ver_b_inc=pi/6000;
h=plot(x,y)
set(h,'MarkerSize',5);
axis([0 3 -0.2 2.8])
grid
theta1=pi/6;
theta2=pi/8;
theta3=pi/8;
theta4=pi/10;
theta5=pi/4.2;
theta6=pi/6;
theta_pel_hor=pi/45;
theta_pel_ver=pi/60;
for k = 1:nframes
   if k<=100
      k;
    theta1=theta1+theta_1_inc;
    theta2=theta2+theta_2_a_inc;
    theta4=theta4+theta_4_a_inc;
    theta5=theta5+theta_5_a_inc;
    theta6=theta6+theta_6_a_inc;
    theta_pel_hor=theta_pel_hor-theta_pel_hor_a_inc;
    theta_pel_ver=theta_pel_ver-theta_pel_ver_a_inc;
    x=[1-lo*sin(theta1),1,1+l1*cos(theta1),1-lo*sin(theta1),1-lo*sin(theta1)-l2*sin(theta2),1-
lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3),1-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-
l_pel*tan(theta_pel_hor),1-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-l_pel*tan(theta_pel_hor)-
l4*sin(theta4),1-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-l_pel*tan(theta_pel_hor)-l4*sin(theta4)-
l5*sin(theta5),1-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-l_pel*tan(theta_pel_hor)-l4*sin(theta4)-
l5*sin(theta5)-loo*sin(theta6),1-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-l_pel*tan(theta_pel_hor)-
l4*sin(theta4)-l5*sin(theta5)-loo*sin(theta6)+l6*cos(theta6),1-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-
l_pel*tan(theta_pel_hor)-l4*sin(theta4)-l5*sin(theta5)];
   y=[lo*cos(theta1),0,l1*sin(theta1),lo*cos(theta1),lo*cos(theta1)+l2*cos(theta2),lo*cos(theta1)+l2
*cos(theta2)+l3*cos(theta3),lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver),lo*cos
(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-
l4*cos(theta4),lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4)-
l5*cos(theta5),lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4)-
l5*cos(theta5)-loo*cos(theta6),lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-
l4*cos(theta4)-l5*cos(theta5)-loo*cos(theta6)-
l6*sin(theta6),lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4)-
l5*cos(theta5)];
   elseif k<=200
      k;
      theta2=theta2+theta_2_b_inc;
      theta3=theta3+theta_3_b_inc;
      theta4=theta4+theta_4_a_inc;
      theta5=theta5+theta_5_b_inc;
       theta6=theta6-1.5*theta_6_a_inc;
       theta_pel_hor=theta_pel_hor-theta_pel_hor_a_inc;
      theta_pel_ver=theta_pel_ver-theta_pel_ver_a_inc;
      x=[1-lo*sin(theta1),1,1+l1*cos(theta1),1-lo*sin(theta1),1-lo*sin(theta1)-l2*sin(theta2),1-
lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3),1-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-
l_pel*tan(theta_pel_hor),1-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-l_pel*tan(theta_pel_hor)-
```

```
l4*sin(theta4),1-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-l_pel*tan(theta_pel_hor)-l4*sin(theta4)-
l5*sin(theta5),1-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-l_pel*tan(theta_pel_hor)-l4*sin(theta4)-
l5*sin(theta5)-loo*sin(theta6),1-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-l_pel*tan(theta_pel_hor)-
l4*sin(theta4)-l5*sin(theta5)-loo*sin(theta6)+l6*cos(theta6),1-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-
l_pel*tan(theta_pel_hor)-l4*sin(theta4)-l5*sin(theta5)];

y=[lo*cos(theta1),0,l1*sin(theta1),lo*cos(theta1),lo*cos(theta1)+l2*cos(theta2),lo*cos(theta1)+l2*cos(thet
a2)+l3*cos(theta3),lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver),lo*cos(theta1)+
l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-
l4*cos(theta4),lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4)-
l5*cos(theta5),lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4)-
l5*cos(theta5)-loo*cos(theta6),lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-
l4*cos(theta4)-l5*cos(theta5)-loo*cos(theta6)-
l6*sin(theta6),lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4)-
l5*cos(theta5)];
            elseif k<=250
             k;
            theta1=theta1+theta_1_inc;
            theta2=theta2+theta_2_c_inc;
            theta3=theta3+theta_2_c_inc;
            theta4=theta4+1.2*theta_4_b_inc;
            theta5=theta5+1.3*theta_5_c_inc;
            theta6=theta6-2*theta_6_a_inc;
            theta_pel_hor=theta_pel_hor-theta_pel_hor_b_inc;
            theta_pel_ver=theta_pel_ver-theta_pel_ver_b_inc;
            x=[1+l1-l1*cos(theta1)-lo*sin(theta1),1+l1-l1*cos(theta1),1+l1,1+l1-l1*cos(theta1)-
lo*sin(theta1),1+l1-l1*cos(theta1)-lo*sin(theta1)-l2*sin(theta2),1+l1-l1*cos(theta1)-lo*sin(theta1)-
l2*sin(theta2)-l3*sin(theta3),1+l1-l1*cos(theta1)-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-
l_pel*tan(theta_pel_hor),1+l1-l1*cos(theta1)-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-
l_pel*tan(theta_pel_hor)-l4*sin(theta4),1+l1-l1*cos(theta1)-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-
l_pel*tan(theta_pel_hor)-l4*sin(theta4)-l5*sin(theta5),1+l1-l1*cos(theta1)-lo*sin(theta1)-l2*sin(theta2)-
l3*sin(theta3)-l_pel*tan(theta_pel_hor)-l4*sin(theta4)-l5*sin(theta5)-loo*sin(theta6),1+l1-l1*cos(theta1)-
lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-l_pel*tan(theta_pel_hor)-l4*sin(theta4)-l5*sin(theta5)-
loo*sin(theta6)+l6*cos(theta6),1+l1-l1*cos(theta1)-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-
l_pel*tan(theta_pel_hor)-l4*sin(theta4)-l5*sin(theta5)];
            y=[-l1*sin(theta1)+lo*cos(theta1),-l1*sin(theta1),0,-l1*sin(theta1)+lo*cos(theta1),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4)-
l5*cos(theta5),-l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-
l4*cos(theta4)-l5*cos(theta5)-loo*cos(theta6),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4)-
l5*cos(theta5)-loo*cos(theta6)-l6*sin(theta6),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4)-
l5*cos(theta5)];
        elseif k<=300
            k
            theta1=theta1+theta_1_inc;
            theta2=theta2+1.67*theta_2_d_inc;
            theta3=theta3-theta_3_d_inc;
            theta4=theta4+3*theta_4_c_inc;
            theta5=theta5+1.43*theta_5_d_inc;
            theta6=theta6-3*theta_6_a_inc;
            theta_pel_hor=theta_pel_hor-theta_pel_hor_b_inc
```

```
            theta_pel_ver=theta_pel_ver-theta_pel_ver_b_inc
            x=[1+l1-l1*cos(theta1)-lo*sin(theta1),1+l1-l1*cos(theta1),1+l1,1+l1-l1*cos(theta1)-
lo*sin(theta1),1+l1-l1*cos(theta1)-lo*sin(theta1)-l2*sin(theta2),1+l1-l1*cos(theta1)-lo*sin(theta1)-
l2*sin(theta2)-l3*sin(theta3),1+l1-l1*cos(theta1)-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-
l_pel*tan(theta_pel_hor),1+l1-l1*cos(theta1)-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-
l_pel*tan(theta_pel_hor)-l4*sin(theta4),1+l1-l1*cos(theta1)-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-
l_pel*tan(theta_pel_hor)-l4*sin(theta4)-l5*sin(theta5),1+l1-l1*cos(theta1)-lo*sin(theta1)-l2*sin(theta2)-
l3*sin(theta3)-l_pel*tan(theta_pel_hor)-l4*sin(theta4)-l5*sin(theta5)-loo*sin(theta6),1+l1-l1*cos(theta1)-
lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-l_pel*tan(theta_pel_hor)-l4*sin(theta4)-l5*sin(theta5)-
loo*sin(theta6)+l6*cos(theta6),1+l1-l1*cos(theta1)-lo*sin(theta1)-l2*sin(theta2)-l3*sin(theta3)-
l_pel*tan(theta_pel_hor)-l4*sin(theta4)-l5*sin(theta5)]
            y=[-l1*sin(theta1)+lo*cos(theta1),-l1*sin(theta1),0,-l1*sin(theta1)+lo*cos(theta1),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4)-
l5*cos(theta5),-l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-
l4*cos(theta4)-l5*cos(theta5)-loo*cos(theta6),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4)-
l5*cos(theta5)-loo*cos(theta6)-l6*sin(theta6),-
l1*sin(theta1)+lo*cos(theta1)+l2*cos(theta2)+l3*cos(theta3)+l_pel*tan(theta_pel_ver)-l4*cos(theta4)-
l5*cos(theta5)]
          elseif k<=360
            k
            theta1_swing=theta1;
            theta2_swing=theta2;
            theta3_swing=theta3;
            theta4_swing=theta4;
            theta5_swing=theta5;
            theta6_swing=theta6
            theta_pel_hor_swing=theta_pel_hor;
            theta_pel_ver_swing=theta_pel_ver;
              theta6_e_inc=pi/450;
              theta5_e_inc=pi/800;
              theta4_e_inc=pi/700;
              theta3_e_inc=pi/450;
              theta2_e_inc=-pi/750;
              theta1_e_inc=-pi/750;
              theta_pel_hor_e_inc=pi/2700;
              theta_pel_ver_e_inc=pi/3600;
              theta_6=theta6_swing+(k-300)*theta6_e_inc;
              theta_5=theta5_swing+(k-300)*theta5_e_inc;
              theta_4=theta4_swing+(k-300)*theta4_e_inc;
              theta_3=theta3_swing+(k-300)*theta3_e_inc;
              theta_2=theta2_swing+(k-300)*theta2_e_inc;
              theta_1=theta1_swing+(k-300)*theta1_e_inc;
              theta_pel_hor_2=theta_pel_hor_swing+(k-300)*theta_pel_hor_e_inc;
              theta_pel_ver_2=theta_pel_ver_swing+(k-300)*theta_pel_ver_e_inc;
              x_s=1+l1-l1*cos(theta1_swing)-lo*sin(theta1_swing)-l2*sin(theta2_swing)-
l3*sin(theta3_swing)-l_pel*tan(theta_pel_hor_swing)-l4*sin(theta4_swing)-l5*sin(theta5_swing)-
loo*sin(theta5_swing)
              y_s=0;

x=[x_s+lo*sin(theta_6),x_s+l6*cos(theta_6),x_s,x_s+lo*sin(theta_6),x_s+lo*sin(theta_6)+l5*sin(theta_5),
x_s+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4),x_s+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)
```

```
+l_pel*tan(theta_pel_hor_2),x_s+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor
_2)+l3*sin(theta_3),x_s+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor_2)+l3*s
in(theta_3)+l2*sin(theta_2),x_s+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor_
2)+l3*sin(theta_3)+l2*sin(theta_2)+lo*sin(theta_1),x_s+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l
_pel*tan(theta_pel_hor_2)+l3*sin(theta_3)+l2*sin(theta_2)+lo*sin(theta_1)+l1*cos(theta_1),x_s+lo*sin(th
eta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor_2)+l3*sin(theta_3)+l2*sin(theta_2)]
            y=[loo*cos(theta_6),-
l6*sin(theta_6),0,loo*cos(theta_6),lo*cos(theta_6)+l5*cos(theta_5),loo*cos(theta_6)+l5*cos(theta_5)+l4*c
os(theta_4),loo*cos(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-
l_pel*tan(theta_pel_ver_2),loo*cos(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-l_pel*tan(theta_pel_ver_2)-
l3*cos(theta_3),loo*cos(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-l_pel*tan(theta_pel_ver_2)-
l3*cos(theta_3)-l2*cos(theta_2),loo*cos(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-
l_pel*tan(theta_pel_ver_2)-l3*cos(theta_3)-l2*cos(theta_2)-
lo*cos(theta_1),loo*cos(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-l_pel*tan(theta_pel_ver_2)-
l3*cos(theta_3)-l2*cos(theta_2)-
lo*cos(theta_1)+l1*sin(theta_1),loo*cos(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-
l_pel*tan(theta_pel_ver_2)-l3*cos(theta_3)-l2*cos(theta_2)]
            elseif k<=420
              k
                theta5_f_inc=pi/1000;
                 theta4_f_inc=pi/1000;
                 theta3_f_inc=pi/300;
                 theta2_f_inc=pi/160;
                 theta1_f_inc=pi/230;
                 theta_pel_hor_f_inc=pi/2700;
                 theta_pel_ver_f_inc=pi/3600;
                 theta_6=theta6_swing+60*theta6_e_inc;
                 theta_5=theta5_swing+60*theta5_e_inc+(k-360)*theta5_f_inc;
                 theta_4=theta4_swing+60*theta4_e_inc+(k-360)*theta4_f_inc
                 theta_3=theta3_swing+60*theta3_e_inc+(k-360)*theta3_f_inc;
                 theta_2=theta2_swing+60*theta2_e_inc+(k-360)*theta2_f_inc;
                 theta_1=theta1_swing+60*theta1_e_inc+(k-360)*theta1_f_inc;
                 theta_pel_hor_2=theta_pel_hor_swing+60*theta_pel_hor_e_inc+(k-
360)*theta_pel_hor_f_inc;
                 theta_pel_ver_2=theta_pel_ver_swing+60*theta_pel_ver_e_inc+(k-
360)*theta_pel_ver_f_inc;

x=[x_s+lo*sin(theta_6),x_s+l6*cos(theta_6),x_s,x_s+lo*sin(theta_6),x_s+lo*sin(theta_6)+l5*sin(theta_5),
x_s+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4),x_s+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)
+l_pel*tan(theta_pel_hor_2),x_s+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor
_2)+l3*sin(theta_3),x_s+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor_2)+l3*s
in(theta_3)+l2*sin(theta_2),x_s+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor_
2)+l3*sin(theta_3)+l2*sin(theta_2)+lo*sin(theta_1),x_s+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l
_pel*tan(theta_pel_hor_2)+l3*sin(theta_3)+l2*sin(theta_2)+lo*sin(theta_1)+l1*cos(theta_1),x_s+lo*sin(th
eta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor_2)+l3*sin(theta_3)+l2*sin(theta_2)]
            y=[loo*cos(theta_6),-
l6*sin(theta_6),0,loo*cos(theta_6),lo*cos(theta_6)+l5*cos(theta_5),loo*cos(theta_6)+l5*cos(theta_5)+l4*c
os(theta_4),loo*cos(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-
l_pel*tan(theta_pel_ver_2),loo*cos(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-l_pel*tan(theta_pel_ver_2)-
l3*cos(theta_3),loo*cos(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-l_pel*tan(theta_pel_ver_2)-
l3*cos(theta_3)-l2*cos(theta_2),loo*cos(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-
l_pel*tan(theta_pel_ver_2)-l3*cos(theta_3)-l2*cos(theta_2)-
lo*cos(theta_1),loo*cos(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-l_pel*tan(theta_pel_ver_2)-
l3*cos(theta_3)-l2*cos(theta_2)-
lo*cos(theta_1)+l1*sin(theta_1),loo*cos(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-
l_pel*tan(theta_pel_ver_2)-l3*cos(theta_3)-l2*cos(theta_2)]
```

```
else k<=480
    k
      theta6_g_inc=-pi/350;
      theta5_g_inc=pi/303;
      theta4_g_inc=pi/1200;
      theta3_g_inc=-pi/500;
      theta2_g_inc=pi/1800;
      theta1_g_inc=pi/570;
      theta_pel_hor_g_inc=pi/2700;
      theta_pel_ver_g_inc=pi/3600;
      theta_6=theta6_swing+60*theta6_e_inc+(k-420)*theta6_g_inc;
      theta_5=theta5_swing+60*theta5_e_inc+60*theta5_f_inc+(k-420)*theta5_g_inc;
      theta_4=theta4_swing+60*theta4_e_inc+60*theta4_f_inc+(k-420)*theta4_g_inc
      theta_3=theta3_swing+60*theta3_e_inc+60*theta3_f_inc+(k-420)*theta3_g_inc;
      theta_2=theta2_swing+60*theta2_e_inc+60*theta2_f_inc+(k-420)*theta2_g_inc;
      theta_1=theta1_swing+60*theta1_e_inc+60*theta1_f_inc+(k-420)*theta1_g_inc;
      theta_pel_hor_2=theta_pel_hor_swing+60*theta_pel_hor_e_inc+60*theta_pel_hor_f_inc;
      theta_pel_ver_2=theta_pel_ver_swing+60*theta_pel_ver_e_inc+60*theta_pel_ver_f_inc;

    x=[x_s+l6-lo*sin(theta_6)-l6*cos(theta_6),x_s+l6,x_s+l6-l6*cos(theta_6),x_s+l6-
l6*cos(theta_6)-lo*sin(theta_6),x_s+l6-l6*cos(theta_6)+lo*sin(theta_6)+l5*sin(theta_5),x_s+l6-
l6*cos(theta_6)+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4),x_s+l6-
l6*cos(theta_6)+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor_2),x_s+l6-
l6*cos(theta_6)+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor_2)+l3*sin(theta
_3),x_s+l6-
l6*cos(theta_6)+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor_2)+l3*sin(theta
_3)+l2*sin(theta_2),x_s+l6-
l6*cos(theta_6)+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor_2)+l3*sin(theta
_3)+l2*sin(theta_2)+lo*sin(theta_1),x_s+l6-
l6*cos(theta_6)+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor_2)+l3*sin(theta
_3)+l2*sin(theta_2)+lo*sin(theta_1)+l1*cos(theta_1),x_s+l6-
l6*cos(theta_6)+lo*sin(theta_6)+l5*sin(theta_5)+l4*sin(theta_4)+l_pel*tan(theta_pel_hor_2)+l3*sin(theta
_3)+l2*sin(theta_2)]

     y=[lo*cos(theta_6)-l6*sin(theta_6),0,-l6*sin(theta_6),lo*cos(theta_6)-
l6*sin(theta_6),lo*cos(theta_6)-l6*sin(theta_6)+l5*cos(theta_5),lo*cos(theta_6)-
l6*sin(theta_6)+l5*cos(theta_5)+l4*cos(theta_4),lo*cos(theta_6)-
l6*sin(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-l_pel*tan(theta_pel_ver_2),lo*cos(theta_6)-
l6*sin(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-l_pel*tan(theta_pel_ver_2)-
l3*cos(theta_3),lo*cos(theta_6)-l6*sin(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-
l_pel*tan(theta_pel_ver_2)-l3*cos(theta_3)-l2*cos(theta_2),lo*cos(theta_6)-
l6*sin(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-l_pel*tan(theta_pel_ver_2)-l3*cos(theta_3)-
l2*cos(theta_2)-loo*cos(theta_1),lo*cos(theta_6)-l6*sin(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-
l_pel*tan(theta_pel_ver_2)-l3*cos(theta_3)-l2*cos(theta_2)-
lo*cos(theta_1)+l1*sin(theta_1),lo*cos(theta_6)-l6*sin(theta_6)+l5*cos(theta_5)+l4*cos(theta_4)-
l_pel*tan(theta_pel_ver_2)-l3*cos(theta_3)-l2*cos(theta_2)]
      end
    set(h,'XData',x,'YData',y)
    M(k) = getframe;
  end
end
%  movie(M)
```

APPENDIX D

This Appendix contains the LISP program by Professor Robert McGhee that was discussed in Chapter V. (Comments are highlighted in green.)

```
;C:\\Documents and Settings\\mcghee\\My Documents\\MARG Simulation\\sagittal-foot-kinematics.cl"

;This code was written in Allegro ANSI Common Lisp, Version 6.2, by Prof.
;Robert B. McGhee (mcghee@montereybay.com) at the Naval Postgraduate School in Monterey,
;CA. Date of last revision: 25 April, 2005.

;x is forward in plane of progression. y is up.

(load "C:/documents and settings/yun/desktop/nps academic/02_2005/moves_4472/final_project/n-link-iterative-dynamics.cl")
(defclass biped-foot (planar-link)
  ((heel-to-ankle-length :accessor heel-to-ankle-length :initform 12) ;All lengths in cm.
   (sole-length :accessor sole-length :initform 24) ;Measured from back of heel to start of big toe.
   (sole-to-ankle-angle :accessor sole-to-ankle-angle :initform (/ pi 3)) ;Positive counterclockwise.
   (sole-angle :accessor sole-angle) ;Foot flat is 0 radians. Toe up is positive.
   (stride-length :accessor stride-length) ;From heel strike to next heel strike on same foot.
   (cycle-time :accessor cycle-time) ;For one full stride.
   (support-state :accessor support-state) ;Values are: swing, heel, sole, toe.
   (heel-position :accessor heel-position) ; '(x y)
   (toe-position :accessor toe-position)
   (ankle-position :accessor ankle-position)
   (waypoint-list :accessor waypoint-list)
   (time-stamp :accessor time-stamp :initform 0)))
(defclass waypoint (biped-foot)
  ((waypoint-name :accessor waypoint-name)
   (waypoint-duration :accessor waypoint-duration) ;Normalized to one second cycle time.
   (rotation-reference-point :accessor rotation-reference-point))) ;To next waypoint.
(defmethod make-waypoint-list ((foot biped-foot) number-of-waypoints)
  (do* ((list nil (cons (make-instance 'waypoint) list))
        (count 0 (1+ count)))
       ((= count number-of-waypoints) (setf (waypoint-list foot) list))))
(defmethod initialize-waypoint ((foot biped-foot) index name duration support reference
                                sole heel toe ankle)
  (let* ((list (waypoint-list foot)) (waypoint (elt list index)) (wp waypoint))
        (setf (waypoint-name wp) name (waypoint-duration wp) duration (support-state wp) support
              (rotation-reference-point wp) reference (sole-angle wp) sole (heel-position wp) heel
          (heel-position wp) heel (toe-position wp) toe (ankle-position wp) ankle)))
(defmethod update-waypoint-list ((foot biped-foot) stride-increment)
  (let* ((list (waypoint-list foot)) (waypoint (first list))
         (new-waypoint (update-waypoint waypoint stride-increment))
         (new-list (append (rest list) (list new-waypoint))))
        (setf (waypoint-list foot) new-list)))
(defmethod update-waypoint ((waypoint biped-foot) stride-increment)
  (setf (heel-position waypoint) (vector-add stride-increment (heel-position waypoint))
        (toe-position waypoint) (vector-add stride-increment (toe-position waypoint))
        (ankle-position waypoint) (vector-add stride-increment (ankle-position waypoint)))
```

```
  waypoint)

(defmethod update-node-positions ((foot biped-foot))
  (let* ((support-state (support-state foot)))
        (case support-state
              ((heel sole)  (update-node-positions-from-heel foot))
              (toe     (update-node-positions-from-toe foot))
              (swing (update-node-positions-from-ankle foot)))))
(defmethod update-node-positions-from-heel ((foot biped-foot)) ;Position is x-y list in x-y plotter
  (let* ((sole-angle (sole-angle foot)) (sl (sole-length foot))  ;coordinates.
        (heel-to-ankle-angle (+ sole-angle (sole-to-ankle-angle foot)))
        (heel-position (heel-position foot)) (htal (heel-to-ankle-length foot))
        (ankle-x-position (+ (first heel-position) (* (cos heel-to-ankle-angle) htal)))
        (ankle-y-position (+ (second heel-position) (* (sin heel-to-ankle-angle) htal)))
        (toe-x-position (+ (first heel-position) (* (cos sole-angle) sl)))
        (toe-y-position (+ (second heel-position) (* (sin sole-angle) sl))))
       (setf (ankle-position foot) (list ankle-x-position ankle-y-position)
             (toe-position foot) (list toe-x-position toe-y-position))))
(defmethod update-node-positions-from-toe ((foot biped-foot));Position is in x-y plotter coordi-
nates.
  (let* ((sole-angle (sole-angle foot)) (sl (sole-length foot))
        (heel-to-ankle-angle (+ sole-angle (sole-to-ankle-angle foot)))
        (toe-position (toe-position foot)) (htal (heel-to-ankle-length foot))
        (heel-x-position (- (first toe-position) (* (cos sole-angle) sl)))
        (heel-y-position (- (second toe-position) (* (sin sole-angle) sl)))
        (ankle-x-position (+ heel-x-position (* (cos heel-to-ankle-angle) htal)))
        (ankle-y-position (+ heel-y-position (* (sin heel-to-ankle-angle) htal))))
       (setf (ankle-position foot) (list ankle-x-position ankle-y-position)
          (heel-position foot) (list heel-x-position heel-y-position))))
(defmethod update-node-positions-from-ankle ((foot biped-foot)) ;Position is in x-y plotter
  (let* ((sole-angle (sole-angle foot)) (sl (sole-length foot)) ;coordinates.
        (heel-to-ankle-angle (+ sole-angle (sole-to-ankle-angle foot)))
        (ankle-position (ankle-position foot)) (htal (heel-to-ankle-length foot))
        (heel-x-position (- (first ankle-position) (* (cos heel-to-ankle-angle) htal)))
        (heel-y-position (- (second ankle-position) (* (sin heel-to-ankle-angle) htal)))
        (toe-x-position (+ heel-x-position (* (cos sole-angle) sl)))
        (toe-y-position (+ heel-y-position (* (sin sole-angle) sl))))
       (setf (toe-position foot) (list toe-x-position toe-y-position)
             (heel-position foot) (list heel-x-position heel-y-position))))
(defmethod draw-foot ((foot biped-foot) (recorder x-y-recorder) line-width scale) ; Does not draw
toes.
   (let* ((heel (heel-position foot)) (ankle (ankle-position foot)) (toe (toe-position foot)))
         (and (draw-wide-line-in-window recorder line-width scale scale ankle heel)
              (draw-wide-line-in-window recorder line-width scale scale heel toe)
              (draw-wide-line-in-window recorder line-width scale scale toe ankle))))
(defun start ()
  (setf right-foot (make-instance 'biped-foot))
  (make-waypoint-list right-foot 6)
  (initialize-waypoint right-foot 0 'heel-strike .1 'heel 'heel .4 '(0 0) nil nil)
  (update-node-positions (elt (waypoint-list right-foot) 0))
  (initialize-waypoint right-foot 1 'foot-flat .5 'heel 'heel .0 '(0 0) nil nil)
  (update-node-positions (elt (waypoint-list right-foot) 1))
  (initialize-waypoint right-foot 2 'toe-off .1 'toe 'toe -.7 nil '(24 0) nil)
  (update-node-positions (elt (waypoint-list right-foot) 2))
  (initialize-waypoint right-foot 3 'early-swing .1 'swing 'ankle -.1 nil nil '(40 15))
  (update-node-positions (elt (waypoint-list right-foot) 3))
```

```
  (initialize-waypoint right-foot 4 'mid-swing .1 'swing 'ankle 0 nil nil '(80 15))
  (update-node-positions (elt (waypoint-list right-foot) 4))
  (initialize-waypoint right-foot 5 'late-swing .1 'swing 'ankle .4 nil nil '(120 15))
  (update-node-positions (elt (waypoint-list right-foot) 5))
  (setf recorder-1 (make-instance 'x-y-recorder))
  (initialize-recorder recorder-1 900 600 '(90 500))
  (draw-coordinate-axes recorder-1)))
(defun tf1 (index) (draw-foot (elt (waypoint-list right-foot) index) recorder-1 3 2))
(defun tf2 () (draw-foot (elt (waypoint-list right-foot) 0) recorder-1 3 2)
          (update-waypoint-list right-foot '(120 0)))
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E

This Appendix presents the Matlab simulation program that processes the real data. This program was discussed in Chapter VI. (Comments are highlighted in green.)

```
% The first part of the code is the "Factored Quaternion Algorithm",
% written by Conrado Aparicio, and receives raw data from the sensors and transforms it first to quaternions
% and then to angles. The second part of the code is written by Pantazis
% Ioannis and implements the data to a simulation which represents the
% motion of the lower extremities.

% PART I - written by Conrado Aparicio - September 2004

format long
close all,
clear all,clc
q_opt=[];
R_cal=[];

% Kalman Filter initialization
% % ==================================
qhat=[];
delta_t = 0.01;
tao1=0.5;
tao2=0.5;
tao3=0.5;
D1=50;
D2=50;
D3=50;
sigma1=sqrt(.005);
sigma2=sqrt(.005);
sigma3=sqrt(.005);
sigma4=sqrt(.0001);%sqrt(1.42e-4);
sigma5=sqrt(.0001);%sqrt(1.42e-4);%1e-6;%sqrt(5.69e-5);
sigma6=sqrt(.0001);%sqrt(1.42e-4);%sqrt(4.03e-5);
sigma7=sqrt(.0001);%sqrt(1.42e-4);%1e-6;%sqrt(1.18e-4);
q11 = (D1/(2*tao1))*(1-exp(-2*delta_t/tao1));
q22 = (D2/(2*tao2))*(1-exp(-2*delta_t/tao2));
q33 = (D3/(2*tao3))*(1-exp(-2*delta_t/tao3));
Q = [q11  0   0  0 0 0 0;
     0  q22  0  0 0 0 0;
     0   0  q33 0 0 0 0;
     0   0   0  0 0 0 0;
     0   0   0  0 0 0 0;
     0   0   0  0 0 0 0;
     0   0   0  0 0 0 0];
R_k = [sigma1^2    0       0       0       0       0       0;
         0     sigma2^2    0       0       0       0       0;
         0       0     sigma3^2    0       0       0       0;
         0       0       0     sigma4^2    0       0       0;
         0       0       0       0     sigma5^2    0       0;
```

```
        0       0       0       0       0     sigma6^2    0;
        0       0       0       0       0       0     sigma7^2];

    phi_1 = [exp(-delta_t/tao1)       0                0;
                0          exp(-delta_t/tao2)       0;
                0                0          exp(-delta_t/tao3)];
    phi_2 = zeros(3,4);
    I = eye(7);
    H = I;
    P = randn(7,1);
    P_k_minus = 10000*I;%diag(P.*P);
    xhat_minus=[];
    xhat_minus(:,1)=[0 0 0 0.5 0.5 0.5 0.5]';

    % Load Sensor calibrating values
    Sensor04

    %Load Sensor Measurements
    load rightthighfullstep.txt % randomArmMotion.txt% NED05.txt%
MargData05aboutZ_CIU3.txt% superfast.txt% NEDpitchUp.txt%              %rotation about x-axis at a rate
of 60 degrees/sec
    MargData = rightthighfullstep; % randomArmMotion;%  NED05;% MargData05aboutZ_CIU3;%
superfast;% NEDpitchUp;%

    %Get reference Magnetic measurement for NED orientation of the sensor
    global Mref;        % magnetic ref
    Mref=[MagRef(1);MagRef(2)]; %Get only x and y components for the factored quaternion algo-
rithm
    Mref=Mref/norm(Mref);
    [aa bb]=size(MargData);
    MargData=MargData(:,2:bb);
    M=MargData(1:aa,9:11)/4096*3;
    A=MargData(1:aa,6:8)/4096*3;
    R=MargData(1:aa,3:5)/4096*3;

    % Data Calibration
    [row colum]=size(M);
    M_cal = (M-repmat(magnull,row,1)).*repmat(magscale,row,1);
    A_cal = (A-repmat(accelnull,row,1)).*repmat(accelscale,row,1);
    tau = 10;
    alpha = 1-(delta_t/tau);
    HPG = (1+alpha)/(2*alpha);
    qhat_minus = [.5 .5 .5 .5]'; % it hs to be a unit quaternion
    rr=[];
    for k=1:aa
       ratenull = (alpha*ratenull) + (1-alpha)*R(k,:);rr(k,:) = ratenull;
       R_filt = HPG * (R(k,:) - ratenull).*ratescale;
       R_cal = [ R_cal ; R_filt ];
       q_factt=fact_quat3([M_cal(k,:)' A_cal(k,:)']); % returns the factored quaternion [q0 q1 q2 q3]
    %     q_factt=Quest([M_cal(k,:)' A_cal(k,:)']);
       q_fact = q_factt;
       error=q_fact-qhat_minus;
       if norm(error) > 1.9
          q_fact=-q_fact;
       end
       q_opt=[q_opt q_factt];
```

```
            % %  Kalman filter
           omega = R_cal(k,:);
            z_k = [omega'; q_fact];
               K_k = P_k_minus*inv( P_k_minus + R_k);
               xhat(:,k) = xhat_minus(:,k) + K_k*(z_k - (xhat_minus(:,k)));
            xhat(4:7,k) = xhat(4:7,k)/norm(xhat(4:7,k));                % Normalize the quaternion estimate
            P_k = (I - K_k)*P_k_minus*[(I - K_k)]' + K_k*R_k*[K_k]';
            p(k) = trace(P_k);
            pm(k) = trace(P_k_minus);
            xhat_minus(1,k+1) = xhat(1,k) + delta_t*(-1/tao1)*xhat(1,k);
            xhat_minus(2,k+1) = xhat(2,k) + delta_t*(-1/tao2)*xhat(2,k);
            xhat_minus(3,k+1) = xhat(3,k) + delta_t*(-1/tao3)*xhat(3,k);
            xhat_minus(4,k+1) = xhat(4,k) + delta_t*(-
0.5)*(xhat(1,k)*xhat(5,k)+xhat(2,k)*xhat(6,k)+xhat(3,k)*xhat(7,k));
            xhat_minus(5,k+1) = xhat(5,k) + delta_t*(0.5)*(xhat(1,k)*xhat(4,k)-
xhat(2,k)*xhat(7,k)+xhat(3,k)*xhat(6,k));
            xhat_minus(6,k+1) = xhat(6,k) + delta_t*(0.5)*(xhat(1,k)*xhat(7,k)+xhat(2,k)*xhat(4,k)-
xhat(3,k)*xhat(5,k));
            xhat_minus(7,k+1) = xhat(7,k) + delta_t*(0.5)*(-
xhat(1,k)*xhat(6,k)+xhat(2,k)*xhat(5,k)+xhat(3,k)*xhat(4,k));
            xhat_minus(4:7,k+1) = xhat_minus(4:7,k+1)/norm(xhat_minus(4:7,k+1));          %Normalize
the quaternion projection
            qhat_minus = xhat_minus(4:7,k+1);
            phi_k = [                          phi_1,                                                  phi_2
;
                  [-xhat(5,k)*0.5*delta_t -xhat(6,k)*0.5*delta_t -xhat(7,k)*0.5*delta_t        1          -
xhat(1,k)*0.5*delta_t -xhat(2,k)*0.5*delta_t -xhat(3,k)*0.5*delta_t;
          xhat(4,k)*0.5*delta_t  -xhat(7,k)*0.5*delta_t  xhat(6,k)*0.5*delta_t xhat(1,k)*0.5*delta_t         1
xhat(3,k)*0.5*delta_t  -xhat(2,k)*0.5*delta_t;
          xhat(7,k)*0.5*delta_t  xhat(4,k)*0.5*delta_t -xhat(5,k)*0.5*delta_t xhat(2,k)*0.5*delta_t -
xhat(3,k)*0.5*delta_t         1            xhat(1,k)*0.5*delta_t;
          -xhat(6,k)*0.5*delta_t xhat(5,k)*0.5*delta_t  xhat(4,k)*0.5*delta_t xhat(3,k)*0.5*delta_t
xhat(2,k)*0.5*delta_t  -xhat(1,k)*0.5*delta_t         1         ]];
            P_k_minus = phi_k*P_k*phi_k' + Q;P_k_minus=diag(diag(P_k_minus));
            qhatt = [xhat(4:7,k)]; qhatt=qhatt/norm(qhatt); % quaternion estimate
            qhat = [qhat qhatt];%_minus];
         end
         time=1:row;
         q_opt_time=[time' q_opt']; %each column has 4 elements, the quaternion. each row has as many
columns as the number of samples we get from each sensor
         qhat_time =[time' qhat'];
         sim('RocketManual2') % computes the euler Angles from the factored quaternion and the euler
angles from the quaternion estimate (qhat)

         % PART II - written by Pantazis Ioannis - May 2005

         EA_thigh=EulerAngles2(1:200,:)*180/pi;
         EA2_thigh=EA_thigh*pi/180;
         [EA_thigh(:,1),EA_thigh(:,2)];
         l1=0.45;                  % length of thigh
         l2=.4;                    % length of shank
         l3=l1;                    % length of supporting thigh
         l4=l2;                    % length of supporting shank
```

```
% insert the measurements concerning the shank
EA_shank1=[ones(105,1);(-1)*ones(18,1);ones(77,1)];
EA_shank21=[76.29444516769182;76.25898709034478;76.19487546315098;76.16876598615809;76.14701087758472;76.13205184695561;76.08909707746278;76.06615297843281;76.03107721602918;76.00267088831822;75.97805117617808;75.95213134933447;
75.91059679089688;75.87726622439567;75.86125095438634;75.83879437991014;75.82728928190740;75.81576825557694;75.77503318548564;75.71799423030409;75.70038450334236;75.67932940215937;75.65497485953675;75.63907071723236;75.61820740004332;
75.60402943811820;75.60977076906335;75.61980139659842;75.62769201770324;75.60215360164034;75.61272228409111;75.65705450258581;75.73709960389024;75.84696375579564;75.99706132652270;76.16880028674828;76.36437252029360;76.56721508909990;
76.76406981507859;76.93718130314373;77.09213123092513;77.27021122229672;77.46998059975337;77.72461968115319;78.01255051367656;78.34619835170764;78.65092161335826;78.91006842837263;79.08673939683688;79.12945609917937;79.00986724369764;
78.74654780718427;78.30149067710207;77.69181885649395;76.90853376704095;76.07384040857325;75.17066790220079;74.19519181917286;73.15244801705703;71.87516879878763;70.34436647516506;68.50711763456638;66.55390679423755;64.71960684976156;
63.21278178901569;62.38973132589867;61.95104985315764;61.57018618744907;61.45348320996450;61.60108340530854;61.95084572445634;62.45850297697209;63.07100844330289;63.74251345087574;64.51482663057500;65.35736888059573;66.28468290898759;
67.25003524563567;68.26853240169430;69.34264140526921;70.50410821702026;71.65619565232599;72.89615799729131;74.15313543801167;75.50669911666596;76.83196460070100;78.12657667921566;79.42018012292473;80.66888844115651;81.80751424040940;
82.87164719398398;83.81163250414097;84.75524265117964;85.58978389448480;86.36834087159643;87.04082931611669;87.59387353991258;88.03799362358836;88.37885989781506;88.59344325116685;88.68307925047637;88.71080438393749;88.71140944157961;
88.68902571636906;88.67943314700281;88.69079008593859;88.71144487135848;88.74052266313946;88.72628711327272;88.71500158853635;88.72796933630312;88.75942505068528;88.84220965010768;88.95281543322396;89.07326517809332;89.17204159030709;
89.19756727872201;89.23596799853451;89.35070400569320;89.58058055957454;89.78898070994397;90.00000000000000;89.66468221480379;89.31286494253048;89.15089648109229;88.97997926202881;88.00004656325169;86.82758351286529;85.89528548240612;
84.67784626560631;83.42881197808107;82.40439516283183;81.50417450888546;80.57792003934385;79.63507571898184;78.75645077084195;77.98360182643883;77.31397224651268;76.67115283043033;76.09449586891236;75.64312935767900;75.25382425404361];

EA_shank22=(10*ones(58,1))+[74.96762268464075;74.74227295509245;74.54046719433117;74.39666956739625;74.30680597040568;74.21179897718174;74.14711789261244;74.09349472159271;74.05767719222916;74.05269364692163;74.08319409250444;74.19124679040114;74.32541353374622;
74.44345014544456;74.55925551374570;74.66319285288451;74.76909825703933;74.87415572829241;74.96395552274591;75.02232006152686;75.08849695280063;75.14764941183837;75.19732834540049;75.30184068016054;75.37452777998756;75.43375807847781;
75.49222669446675;75.55296515055912;75.65962314352377;75.76468104079719;75.85264888313724;75.94465649824930;76.02718460913641;76.08757573468162;76.15717309050814;76.23473164059698;76.28381277471297;76.30466000158124;76.34851845159211;
76.40559518259376;76.50122738792454;76.60817647591622;76.69975158472387;76.79174503757280;76.88188601883421;76.94050214480657;77.00004525752304;77.07020594886336;77.12833983301229;77.19800263848271;77.26080170146199;77.32308854608472;
77.37309373056630;77.41809247823214;77.4485180762838;77.49064355867631;77.52327485359959;77.56159751733415];
EA_shank2=[EA_shank21;EA_shank22]
EA_shank=[EA_shank1,EA_shank2]
EA2_shank=EA_shank*pi/180;

% insert the measurements concerning the supporting thigh
EA_sup_thigh1=[(-1)*ones(92,1);ones(108,1)];
```

```
EA_sup_thigh2=100*[0.81208896180412;0.81262339712807;0.81265493315408;0.81220913500
677;0.81144036155615;0.81071808573534;0.80981962017485;0.80880359393591;0.80746761446377;0.8
0594448014483;0.80442702880745;0.80270465682224;0.80111690906816;
0.79984076916483;0.79882146288581;0.79815014518627;0.79802944365901;0.79820478945139;0.7985
1162254753;0.79893172330094;0.79961674445157;0.80051976481299;0.80150234479199;0.8024656448
8709;0.80340540450625;0.80425508353976;0.80580189842780;
0.80753474637967;0.80931297206253;0.81211046676479;0.81555466830462;0.81940528492108;0.8237
6691757932;0.82869266734373;0.83375530376260;0.83899038830318;0.84410789638780;0.8487261986
2057;0.85277684755907;0.85622162730903;0.85871539566073;
0.86038767525222;0.86185202656324;0.86316863370845;0.86451060963610;0.86575420898734;0.8673
7408650786;0.86888288519753;0.87070124445594;0.87279870934512;0.87504541097800;0.8770252367
5087;0.87875512737918;0.88019044010944;0.88104368162488;
0.88205093560789;0.88248683828116;0.88277389877805;0.88242912320257;0.88185701610579;0.8810
3720481551;0.88032677328632;0.87951421975067;0.87867554226063;0.87782768514641;0.8769670672
1120;0.87587949036489;0.87509644584962;0.87385458102020;
0.87302842995390;0.87213672570005;0.87077174012203;0.86979077655528;0.86859916583152;0.8681
0459136526;0.86825226573316;0.86843746482671;0.86855047479725;0.86906305896723;0.8692153538
2112;0.86918071520107;0.86884665552536;0.86900120831288;
0.86925168007305;0.86929640568779;0.86964182201518;0.87044302084752;0.87141720220391;0.8726
7432614132;0.87435438476143;0.87623540651941;0.87869975980009;0.88118829595354;0.8839498458
3836;0.88728149177098;0.89055789725341;0.89270519709689;
0.89444222469540;0.89636057649710;0.89851608472366;0.90000000000000;0.89939121446693;0.8989
2601342380;0.89800919721537;0.89679477445422;0.89708397177871;0.89642727736140;0.8943684459
3802;0.89198094539749;0.88938727775713;0.88644589843121;
0.88319750036185;0.87947390717692;0.87626149993330;0.87390751778260;0.87206672491652;0.8703
1103075981;0.86876227978168;0.86708503150964;0.86506574286288;0.86296036712181;0.8610720759
6335;0.85930153615555;0.85777447793126;0.85615656328293;
0.85444275743586;0.85315761487225;0.85215606216799;0.85113493415425;0.85025418108227;0.8495
8482881612;0.84891694999846;0.84846319848197;0.84796746717649;0.84736191536801;0.8466391061
4098;0.84620822651993;0.84581560213136;0.84564420135650;
0.84581010143275;0.84605281098612;0.84631737536776;0.84680659651290;0.84742620126069;0.8479
5808905599;0.84881535117469;0.84957736632996;0.85052839588649;0.85139696244070;0.8524914194
0626;0.85333100890564;0.85409963602872;0.85477095819955;
0.85528398701616;0.85600758560993;0.85698615831678;0.85791360989160;0.85914298484313;0.8605
2453397959;0.86212761551655;0.86347395888432;0.86501772736091;0.86662865223590;0.8681480864
2568;0.86963028359508;0.87099725343070;0.87228946640297;
0.87350983481403;0.87441944878053;0.87505588410618;0.87545194811810;0.87566591379513;0.8757
8281060499;0.87604152365413;0.87607662167363;0.87619052107411;0.87603187920816;0.8760168499
1591;0.87587268088960;0.87570092494379;0.87561897143800;
0.87579538495010;0.87596555531847;0.87623701130239;0.87648541605913;0.87679992384545;0.8767
4722565011;0.87673112666920;0.87668087740364;0.87662732762323;0.87643765786815;0.8765705875
7300;0.87674385412704;0.87682692108184;0.87701356480309;
0.87728016683646;0.87759209059653;0.87782792992677;0.87809139583122;0.87842377987383];
EA_sup_thigh=[EA_sup_thigh1,EA_sup_thigh2];
EA2_sup_thigh=EA_sup_thigh*pi/180;

% insert the measurements concerning the supporting shank
EA_sup_shank1=ones(200,1);
EA_sup_shank2=100*[0.88145694377428;0.88098600666114;0.88054049448316;0.8802302897
2148;0.87988768544396;0.87940500414682;0.87887611740096;0.87865921072628;0.87857205026314;0.
87821947333709;0.87782657624405;0.87746993606374;0.87708618738439;
0.87704186548039;0.87691270262198;0.87666895151648;0.87653777948691;0.87605515361242;0.8757
1896762354;0.87546495794080;0.87518382769547;0.87483557214028;0.87453106662142;0.8743584010
5168;0.87414318038022;0.87382095179957;0.87344674747826;
0.87304109782893;0.87261302440952;0.87191025206910;0.87137154831996;0.87075279774303;0.8702
4369685157;0.86998134997923;0.86962798904008;0.86901692863969;0.86858837139511;0.8681509317
```

```
3475;0.86764788296317;0.86715009776390;0.86640548508302;
0.86560340739622;0.86509431728253;0.86468125942468;0.86414023146833;0.86369927678767;0.8632
6525314249;0.86273906080225;0.86214363092718;0.86138976018066;0.86055635685959;0.8597764252
9815;0.85919995116320;0.85867984855068;0.85806692604993;
0.85759651131424;0.85729425579491;0.85704651849988;0.8569948459960;0.85695716283171;0.85674
838608400;0.85666172665302;0.85728544277858;0.85805935399425;0.85907543901582;0.86017202032
335;0.86105844168805;0.86138051081723;0.86108216537261;
0.86091398612915;0.86130172323762;0.86127780689981;0.86105054032341;0.86014674306744;0.8582
8637140000;0.85696933153254;0.85708190712982;0.85749302177820;0.85639037733685;0.8543430626
1477;0.85213423206423;0.85040661566885;0.84842491322758;
0.84561626440386;0.84245187363290;0.83880882589258;0.83568951788896;0.83334277804874;0.8315
3074306578;0.82877670244191;0.82572426475620;0.82270846905290;0.81961457664385;0.8165519818
4184;0.81374336616149;0.81138969873123;0.80948031661181;
0.80709141643393;0.80358808779429;0.79969218207858;0.79558293706615;0.79056558748186;0.7849
9695293050;0.77887065592652;0.77273043779037;0.76680354710089;0.76079628838865;0.7548321341
9288;0.74924786554812;0.74409435379501;0.73908834546689;
0.73347812236586;0.72742784258141;0.72015871709967;0.71323493265998;0.70661002901194;0.7005
0093367219;0.69550838040394;0.69214835103944;0.69084899811589;0.69093485813566;0.6916807734
7566;0.69265661642201;0.69247553867062;0.69216121849557;
0.69163244766057;0.68970705257381;0.68532635044989;0.67804382550346;0.67229039660690;0.6691
0070249944;0.66649790163919;0.66391470021779;0.65949068520586;0.65278617118391;0.6433326266
8914;0.63154186100421;0.62103367898254;0.60658423549736;
0.59348625684051;0.58048679020013;0.56806001505859;0.55659095555474;0.54605009249110;0.5360
6403925300;0.52689402564668;0.51821219872584;0.51040399677364;0.50334764949419;0.4972463928
3650;0.49206210872551;0.48789109039820;0.48511330297387;
0.48379757207430;0.48341570456771;0.48387024113309;0.48489905137038;0.48631457015798;0.4879
0971207843;0.48925540665776;0.49084942576778;0.49239520572320;0.49410520935197;0.4954636342
6479;0.49655825033453;0.49728764890669;0.49760375286584;
0.49741313338138;0.49731292799721;0.49724386186680;0.49755672530020;0.49798005476640;0.4984
8808906330;0.49958173907326;0.50097921038093;0.50237081075510;0.50357684339041;0.5047764495
7548;0.50579643078758;0.50689418315434;0.50809417077050;
0.50914482003942;0.51033527195124;0.51149902942117;0.51278426738438;0.51424901296548;0.5154
8049524739;0.51674386311984;0.51797441727474;0.51879752501351;0.51926929315194;0.5196774093
4591;0.52006181128113;0.52065026491371;0.52120862944548;
0.52158568082287;0.52190638988448;0.52209055246057;0.52227347565176;0.52260519079751];
        EA_sup_shank=[EA_sup_shank1,EA_sup_shank2];
        EA2_sup_shank=EA_sup_shank*pi/180;

        % corec=(90-max(EA_thigh(:,2)))*pi/180;
        x0=1;
        y0=0;
        x=[x0,x0-l4*cos(pi-EA2_sup_shank(1,2)),x0-l4*cos(pi-EA2_sup_shank(1,2))-
l3*cos(EA2_sup_thigh(1,2)),x0-l4*cos(pi-EA2_sup_shank(1,2))-l3*cos(EA2_sup_thigh(1,2))-
l1*cos(EA2_thigh(1,2)),x0-l4*cos(pi-EA2_sup_shank(1,2))-l3*cos(EA2_sup_thigh(1,2))-
l1*cos(EA2_thigh(1,2))-l2*cos(EA2_shank(1,2))]
        y=[y0,y0+l4*sin(pi-EA2_sup_shank(1,2)),y0+l4*sin(pi-
EA2_sup_shank(1,2))+l3*sin(EA2_sup_thigh(1,2)),y0+l4*sin(pi-
EA2_sup_shank(1,2))+l3*sin(EA2_sup_thigh(1,2))-l1*sin(EA2_thigh(1,2)),y0+l4*sin(pi-
EA2_sup_shank(1,2))+l3*sin(EA2_sup_thigh(1,2))-l1*sin(EA2_thigh(1,2))-l2*sin(EA2_shank(1,2))]
            nframes=length(EA2_thigh);
         h=plot(x,y)
         set(h,'MarkerSize',5);
         axis([0 2 0 1])
         for k=1:200
        k
        omega=pi-EA2_sup_shank(k,2);
```

```
            if EA_sup_thigh(k,1)>=0
               chi=180-EA_sup_thigh(k,2);
            else
               chi=EA_sup_thigh(k,2);
            end
            if EA_thigh(k,1)>=0
               psi=EA_thigh(k,2);
            else
               psi=180-EA_thigh(k,2);
            end
            if EA_shank(k,1)>=0
               phi=EA_shank(k,2);
            else
               phi=180-EA_shank(k,2);
            end
            x=[x0,x0-l4*cos(pi-EA2_sup_shank(k,2)),x0-l4*cos(pi-EA2_sup_shank(k,2))-
l3*cos(chi*pi/180),x0-l4*cos(pi-EA2_sup_shank(k,2))-l3*cos(EA2_sup_thigh(k,2))-
l1*cos(psi*pi/180),x0-l4*cos(pi-EA2_sup_shank(k,2))-l3*cos(EA2_sup_thigh(k,2))-l1*cos(psi*pi/180)-
l2*cos(phi*pi/180)];
            y=[y0,y0+l4*sin(pi-EA2_sup_shank(k,2)),y0+l4*sin(pi-
EA2_sup_shank(k,2))+l3*sin(chi*pi/180),y0+l4*sin(pi-
EA2_sup_shank(k,2))+l3*sin(EA2_sup_thigh(k,2))-l1*sin(psi*pi/180),y0+l4*sin(pi-
EA2_sup_shank(k,2))+l3*sin(EA2_sup_thigh(k,2))-l1*sin(psi*pi/180)-l2*sin(phi*pi/180)];
             set(h,'XData',x,'YData',y)
                M = getframe;
             x(1,5)
            end
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Eric R. Bachmann, *Inertial and Magnetic Tracking of Limb Segment Orientation for Inserting Humans into Synthetic Environments*, Ph.D. Dissertation, Naval Postgraduate School, Monterey, California, December 2000.

2. Andreas Kavousanos-Kavousanakis, *Design and Implementation of a DSP-Based Control* Interface *Unit (CIU),* Master's Thesis, Naval Postgraduate School, Monterey, California, 2004.

3. Conrado Aparicio, *Implementation of a Quaternion-Based Kalman Filter for Human Body Motion Tracking Using MARG Sensors,* Master's Thesis, Naval Postgraduate School, Monterey, California, 2004.

4. Synthetic Environment Laboratory, *Synthetic Environments,* http://www.sel.bee.qut.edu.au/understand/understand.htm, last accessed May 2005.

5. Cynthia C. Norkin and Pamela K. Levangie, *Joint Structure and Function,* Second Edition, pp. 3-6, 449-462, F. A. Davis Company, Philadelphia, 1992.

6. Alexander R. McNeill, *Modeling Step by Step,* 1997-2004-Millenium Mathematics Project, University of Cambridge, http://plus.maths.org/issue13/features/walking, last accessed November 2004.

7. Ed Ayyappa, *Normal Human Locomotion, Part 1: Basic Concepts and Terminology,* American Academy of Orthotists and Prosthetists, http://www.oandp.org/jpo/library/1997_01_010.asp, last accessed March 2005.

8. David A. Winter, *Biomechanics and Motor Control of Human Movement,* Second Edition, John Wiley & Sons, New York, 1990.

9. J. Neils and J. H. Oakley, *Striving for Excellence: Ancient Greek Childhood and the Olympic Spirit*, Alexander S. Onassis Public Benefit Foundation, New York, http://66.102.7.104/search?q=cache:BjQtVW8SDscJ:www.onassis.gr/greek/assoc/enim_deltio/24_04/p36.html+%CE%B1%CE%BC%CF%86%CE%BF%CF%81%CE%AD%CE%B1%CF%82&hl=en, last accessed March 2005.

10. Site of the Cave of Lascaux, France, the Shaft of the Dead Man, http://www.culture.gouv.fr/culture/arcnat/lascaux/en/, last accessed March 2005.

11. Professional Staff Association, *Observational Gait Analysis Handbook,* Rancho Los Amigos Medical Center, Downey, California, 1989.

12. Jacquelin Perry, *Normal Human Locomotion,* www.smpp.nwu.edu/~jim/kinesiology/partB_GaitMechanics.ppt.pdf, last accessed March 2005.

13. L. E. Larson, P. Odenrick, B. Sandlund, P. Weitz and P.A. Oberg, *The Phases of Stride and their Interaction in Human Gait,* Scandinavian Journal of Rehabilitation, Vol. 12, No. 3, pp. 107–112, 1980.

14. Mehmet Bediz, *A Computer Simulation Study of a Single Rigid Body Dynamic Model for Biped Postural Control,* Master's Thesis, Naval Postgraduate School, Monterey, California, 1997.

15. A. G. Bharatkumar, K. E. Daigle, M. G. Pandy, Qin Cai and J. K. Aggarwal, "Lower Limb Kinematics of Human Walking with the Medial Axis Transformation," *Proceedings of the 1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp.116-123, 1994.

16. John J. Graig, *Introduction to Robotics: Mechanics and Control,* Second Edition, pp. 5-84, Addison-Wesley Publishing Company, Massachusetts, 1989.

17. Jia-Ching Cheng and José M. F. Moura, "Tracking Human Walking in Dynamic Scenes," *ICIP'97, IEEE Proceedings of International Conference on Image Processing***,** Vol. I, pp. 137-140, Santa Barbara, California, October 1997.

18. Jacob Rosen, *Models of Robotic Manipulation,* Biorobotics Laboratory, University of Washington, http://brl.ee.washington.edu/Education/ EE543/ Ee543_2001/EE543_2001.html, last accessed March 2005.

19. Oussama Khatib and Dezhen Song, *Spatial Descriptions and Coordinate Transforms,* Stanford University, Notes for CPSC 452, http://faculty.cs.tamu.edu/ dzsong/teaching/spring2005/cpsc452/452_lecture3.ppt, last accessed March 2005.

20. Louise A. Gilchrist and David A. Winter, "A Multisegment Computer Simulation of Normal Human Gait," *IEEE Transaction of Rehabilitation Engineering,* Vol. 5, No. 4, pp. 290–299, December 1997.

21. M. Y. Zarrugh and C. W. Radcliffe, "Simulation of Swing Phase Dynamics in Above-Knee Prosthesis," *Journal of Biomechanics,* Vol. 9, pp. 283-292, 1976.

22. M. G. Pandy and N. Berme, "Quantitative Assessment of Gait Determinants During Single Stance via a Three-Dimensional Model. Part 1. Normal Gait," *Journal Biomechanics,* Vol. 22, pp. 717-724, 1989.

23. H. Henami, Y. F. Zheng and M. J. Hines, "Initiation of Walk and Tiptoe of a Planar Nine-Link Biped," *Mathematics Bioscience*, Vol. 61, pp. 163-189, 1982.

24. M. Ju and J. M. Mansour, "Simulation of the Double Limb Support Phase of Human Gait," *Journal of Biomechanics Engineering,* Vol. 110, pp. 223-229, 1988.

25. F.M.L. Amirouche, S. K. Ider and J. Trimble, "Analytical Method for the Analysis and Simulation of Human Locomotion," *Journal of Biomechanics Engineering*, Vol. 112, pp. 379-386, 1990.

26. S. Onyshko and D. A. Winter, “A Mathematical Model for the Dynamics of Human Locomotion,” *Journal of Biomechanics,* Vol. 13, pp. 361-368, 1980.

27. M. G. Pandy and N. Berme, “A Numerical Method for Simulating the Dynamics of Human Walking,” *Journal of Biomechanics,* Vol. 21, No. 12, pp. 1043-1051, 1988.

28. M. G. Pandy and N. Berme, “Synthesis of Human Walking: A Planar Model for Single Support,” *Journal of Biomechanics,* Vol. 21, No. 12, pp. 1053-1060, 1988.

29. D. T. Davy and M. L. Audu, “A Dynamic Optimization Technique for Predicting Muscle Forces in the Swing Phase of Gait,” *Journal of Biomechanics,* Vol. 20, No. 2, pp. 187-201, 1987.

30. D. A. Meglan, *Enhanced Analysis of Human Locomotion,* Ph. D. Dissertation, Ohio State University, 1991.

31. Jia-Ching Cheng and Jose M. F. Moura, “Automatic Recognition of Human Walking in Monocular Image Sequences,” *Journal of VLSI Signal Processing Systems for Signal, Image and Video Processing,* pp. 107-120, October 1998.

32. R. McGhee, Notes for MV 4472 (Physics for Game Developers & Virtual Environments), Naval Postgraduate School, 2005 (Unpublished).

33. N. Ichiguchi, D. Komaki, H. Ishii, H. Shimoda and H. Yoshikawa, “Experimental Analysis on Human Walking Animation of Virtual Robot in 3D Visual Space,” *IEEE SMC '99 Conference Proceedings,* Vol. 6, pp. 918-923, Tokyo, 1999.

34. Steven Feiner, Blair McIntyre, Tobias Hollerer, and Anthony Webster, “A Touring Machine: Prototyping 3D Augmented Reality Systems for Exploring the Urban Environment,” *Proceedings of International Symposium on Wearable Computers (ISWC 97),* pp. 74-81, 1997.

35. Masakatsu Kurogi and Takeshi Kurata, “Personal Positioning based on Walking Locomotion Analysis with Self-contained Sensors and a Wearable Camera,” *Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality,* pp. 103-112, IEEE Press, New York, March 2003.

36. Eric R. Bachmann, Xiaoping Yun, and Robert B. McGhee, “Sourceless Tracking of Human Posture Using Small Inertial/Magnetic Sensors,” *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA 2003)*, Vol. 2, pp. 822-829, Kobe, Japan, July 16-20, 2003.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman Code EC
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

4. Chairman Code IS
   Department of Information Science
   Naval Postgraduate School
   Monterey, California

5. Professor Xiaoping Yun, Code EC/Yx
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

6. Professor David C. Jenn, Code EC/Jn
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

7. Professor Robert B. McGhee
   Department of Computer Science, Code CS/Mz
   Naval Postgraduate School
   Monterey, California

8. Professor Eric R. Bachman
   Department of Computer Science and System Analysis
   Miami University
   Oxford, Ohio

9. Pantazis Ioannis
   Athens, Greece